CAAI Transactions on Intelligence Technology

WILEY

ORIGINAL RESEARCH OPEN ACCESS

MAPFL: Modular Adaptive Personalised Federated Learning From Visual Sentiment to Physical Activity

Chang Liu[1] | Yapeng Li[2] | Zengmao Wang[2] | Dewen Soh[3] | Bo Du[2,4]

[1]Information Center, Renmin Hospital of Wuhan University, Wuhan, China | [2]School of Computer Science, Wuhan University, Wuhan, China | [3]Singapore University of Technology and Design, Singapore | [4]Renmin Hospital of Wuhan University, Wuhan, China

Correspondence: Bo Du (dubo@whu.edu.cn) | Zengmao Wang (wangzengmao@whu.edu.cn)

Received: 2 December 2025 | **Revised:** 4 February 2026 | **Accepted:** 1 March 2026

ABSTRACT

Human-centric sensing tasks such as visual sentiment analysis and smartphone-based human activity recognition (HAR) exhibit strong client heterogeneity from feature, label, and concept shifts. Among these, concept shift—how individual users interpret and generate human-centric signals—is particularly challenging yet underexplored in federated learning. We propose MAPFL (Modular Adaptive Personalised Federated Learning), a personalised federated framework that performs encoder-level personalisation while preserving standard communication. Each client model is decomposed into a shared encoder and a local classifier head, and equipped with both global and personalised encoders. A gradient-aligned personalisation controller learns client-specific mixing coefficients that adaptively balance the two encoders based on the alignment of their gradients, allowing each client to decide how much to share versus personalise at the representation level. This design supports both image- and sensor-based tasks. On an affective-computing benchmark built from three in-the-wild visual sentiment datasets (ARTphoto, ABSTRACT, PARA) and on a smartphone-based HAR benchmark derived from UCI HAR, MAPFL improves AUC and accuracy by up to 3.3% over strong federated and personalised baselines and converges faster and more stably. These results show that gradient-aligned encoder personalisation is an effective and practical mechanism for handling multi-dimensional heterogeneity in human-centric sensing.

1 | Introduction

The rapid proliferation of smartphones, wearables, and ambient IoT devices has given rise to a new wave of human-centric sensing, in which intelligent systems continuously capture visual and physical signals from daily life. Typical examples include image-based sentiment and aesthetics assessment on personal photo streams, and sensor-based human activity recognition (HAR) for healthcare, fitness, and safety applications. In parallel, recent advances in deep learning have demonstrated that rich representations learned from large-scale data can greatly enhance recognition and prediction performance in both visual and sensor domains. However, centralising such personal data in the cloud is increasingly undesirable or infeasible due to privacy regulations, bandwidth limitations, and users' expectations of data sovereignty. Federated learning (FL) has thus become an important framework for training shared models directly on distributed devices, where only model parameters or gradients are transmitted to a central server [1, 2]. It has been applied in areas including healthcare [3–5], mobile networks [6], and smart sensing [7], and it is increasingly expected to enable scalable, robust, and fair on-device intelligence [8].

Despite its promise, deploying FL in human-centric sensing is non-trivial. Compared with traditional machine learning settings, human-centric data are inherently personalised, noisy, and heterogeneous. In this work, we focus on two widely

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.

© 2026 The Author(s). *CAAI Transactions on Intelligence Technology* published by John Wiley & Sons Ltd on behalf of The Institution of Engineering and Technology and Chongqing University of Technology.

studied tasks that typify these challenges: image-based sentiment classification (ISC) and human activity recognition (HAR). In ISC, users assign subjective emotional or sentiment labels to their own images. Different users may systematically disagree on the same content, resulting in client-specific label and concept shift in the conditional distribution [9, 10]. In HAR, devices collect multivariate time series from accelerometers and gyroscopes under diverse hardware conditions, sensor placements, and behavioural patterns, causing strong feature and mixture shifts across clients [7, 11]. These two forms of heterogeneity, subjective label shift and sensor-level feature drift, may appear independently in practice or coexist within the same system.

From a system-level perspective, this multi-shift heterogeneity poses tensions among four key desiderata that have been emphasised by recent FL literature [2, 8, 12]. First, *generalisation*: the shared knowledge extracted by the federation should transfer across clients and tasks, avoiding overfitting to any single user or domain. Second, *robustness*: the aggregation process and trained models should remain stable under non-IID data, partial participation, and noisy updates, which are ubiquitous in real deployments [13–15]. Third, while *fairness* is an important system-level consideration in many FL deployments, in this work we mainly focus on improving robustness and client-level personalisation. A detailed treatment of group- or client-level fairness is beyond our current scope and we view fairness-aware objectives and evaluations as a complementary direction for future work. Fourth, *personalisation*: each user or device should be able to adapt the shared model to its own data, particularly when the global model cannot represent all idiosyncratic patterns [12, 16].

While classical FL algorithms such as FedAvg [1], FedProx [13] and SCAFFOLD [14] primarily aim at improving global optimization and robustness, they typically optimise a single global model that may not align with all local distributions under strong non-IID conditions [2]. Conversely, many personalised FL (PFL) methods target individual adaptation but may compromise global robustness or fairness if personalisation is not carefully controlled [12].

A growing body of work addresses personalisation in FL through different mechanisms. One line of research treats PFL as a form of multi-task or meta-learning, jointly learning a set of client-specific models with shared structure [16, 17]. Another line designs personalisation layers, where a subset of network layers (often the final classifier) is kept local while the remaining backbone is shared across clients [7, 18]. This approach is attractive for our context: sharing a universal feature extractor can help generalisation and robustness, while private heads capture user-specific label semantics. However, personalisation-layer methods typically rely on a manually fixed split between global and local layers; the same architecture and split are applied to all clients and all tasks. Such static design can be suboptimal when the degree of heterogeneity varies across users or when moving from visual to sensor modalities. Indeed, adaptive architectural strategies have been explored to better handle such data heterogeneity in federated visual recognition [19, 20]. A third line adopts global–local interpolation: each client maintains a personalised model alongside the global model and forms a mixture via a tunable coefficient, sometimes with theoretical guarantees [21–23]. These methods enjoy a clean formulation but often use a single mixing parameter uniformly across the entire network, limiting their ability to personalise different layers or components to different extents. Recent works further explore prototype-based [24], clustering-based [25], representation-space correction [26], and inter-client collaboration [27] strategies, but most still focus on either global generalisation and robustness or local personalisation, rather than balancing all four desiderata for heterogeneous human-centric tasks.

In parallel, the FL community has increasingly recognized the importance of robustness, privacy, and fairness at the system level. For example, Kang et al. [6] analysed reliability issues in mobile FL deployments; Xu et al. [4] and Liu et al. [5] discussed domain-specific challenges in healthcare federations, including data imbalance and center-specific biases. The recent survey of Huang et al. [8] explicitly frames FL research along the axes of generalisation, robustness, and fairness, calling for methods that can safeguard performance across these dimensions, especially under heterogeneous and potentially adversarial conditions. Several studies have examined federated optimization under practical constraints: dynamic differential privacy for client updates [28], privacy incentives via reverse auctions and negotiations [29], decentralised aggregation with partial gradients [30], and adaptive fairness-aware computing for FL [31]. In edge-assisted mobile IoT, Zhou et al. [32] design efficient device scheduling strategies for FL. Similarly, Zhang et al. [33] propose an embedded vertical-federated feature selection algorithm based on particle swarm optimization to support FL with heterogeneous features. These works emphasise that building deployable FL systems requires not only algorithmic improvements but also careful consideration of communication, privacy, incentives, and resource fairness. However, most of them still assume a single global model and do not directly tackle the fine-grained personalisation required by human-centric sensing.

Against this background, we ask: Can we design a single personalised federated learning framework that remains robust under severe multi-shift heterogeneity and provides effective client-level personalisation across both visual and sensor-based tasks, without per-task architecture redesign or extensive hyper-parameter tuning? To answer this question, we propose MAPFL (Modular Adaptive Personalised Federated Learning), a robust PFL framework tailored for human-centric sensing but generic in architecture and modality. MAPFL is built around three interacting modules:

1.1 | M1: Dual-Path Modular Backbone

MAPFL represents each client model as an encoder–head composition and instantiates two coupled encoder–head copies on every participating client: a global copy that tracks the server-side parameters and a personalised copy that stays on-device. These copies are trained through a dual-path local optimization scheme that updates them on the same mini-batches, so that global knowledge continues to adapt to the federation while personalised trajectories capture client-specific patterns. The encoder–head abstraction accommodates both image encoders such as MobileNetV2 [34] and convolutional

HAR encoders [7, 11] under the same framework, and reflects the intuition that feature extractors and decision boundaries may require different degrees of sharing and personalisation.

1.2 | M2: Gradient-Aligned Mixture Controller

On top of this backbone, MAPFL introduces client-specific mixing coefficients $(\alpha_{b,i}, \alpha_{h,i})$ that interpolate between global and personalised parameters at the encoder and head modules. These coefficients are updated using gradient-aligned surrogate gradients inspired by mixture-based PFL methods [21]: when local and global gradients are aligned, the controller increases the global share; when they diverge, it leans towards the personalised parameters. This provides a light-weight, data-driven way to decide how much each module should be shared.

1.3 | M3: Plateau-Scheduled Decoupling

To stabilise training and allow encoder and head to personalise to different extents, MAPFL equips each client with a plateau detector that monitors the personalised-path loss. Early in training, the controller operates in a coupled regime, with $\alpha_{b,i} \approx \alpha_{h,i}$ behaving like a single APFL-style mixing weight; once improvement saturates, the detector triggers a decoupled phase in which encoder and head coefficients are updated separately. This two-stage schedule yields stable convergence at the beginning while enabling richer module-specific personalisation in later rounds.

1.4 | A Note on Privacy

MAPFL keeps the personalised copy v_i (including the personalised head) and the mixing coefficients on-device; the server only receives the updated global copy w_i following the standard FedAvg-style communication. While we do not provide a rigorous privacy analysis or formal privacy guarantees in this work, avoiding transmission of client-specific classifier heads (which are tightly coupled to labels) may reduce the exposure of label-related information compared to approaches that communicate personalised classifiers.

By combining these components, MAPFL provides a modular umbrella over several existing PFL design patterns while making their relationships explicit. When the mixing coefficients are driven towards zero and the personalised copy is ignored, MAPFL collapses to a standard global FL model in the spirit of FedAvg. When the coefficients favour the personalised encoder and head and the global copy is mainly used for initialisation, the framework behaves similarly to layer-wise personalisation and head-tuning approaches such as FedPer or FedBABU, where most capacity is kept local. When both encoder and head coefficients remain strictly between 0 and 1, the resulting behaviour is related to mixture-based PFL methods such as APFL, FedALA, and GPFL [12, 22, 23], but with two key differences: MAPFL applies mixing at the encoder–head module level rather than on the whole network, and it updates the mixing coefficients using the gradient-aligned surrogate of Section 3.3 instead of an additional meta-optimization or bi-level inner loop. Unlike prior work, MAPFL is explicitly designed to operate across heterogeneous modalities (images and time-series), and to be evaluated under the lens of robustness and personalisation as advocated in recent FL surveys [2, 8]. In addition, MAPFL is compatible with recent software efforts such as PFLlib [35], which provide standardized benchmarks and implementations for personalised FL, making it easier to compare algorithms in a unified experimental environment.

We validate MAPFL on two complementary case studies. The first is *visual sentiment analysis*, where we model image-level labels on datasets such as PARA [10] and affective image collections derived from ARTphoto [36]. These benchmarks capture user-perceived sentiment and aesthetics from in-the-wild images, and are closely related to federated image emotion analysis tasks such as CAFL [9]. The second is *smartphone-based human activity recognition*, built upon the UCI HAR dataset [11] and variants used in federated HAR frameworks [7]. This setting exposes feature-level heterogeneity due to user and device differences, but labels are relatively objective compared to ISC. MAPFL is evaluated against strong FL and PFL baselines including FedAvg [1], FedProx [13], SCAFFOLD [14], FedBN [37], MOON [38], FedNTD [39], FedLC [40], FedBABU [41], FedDBE [26], prototype- and representation-based PFL methods [22–24], as well as the recent class-wise federated averaging method cwFedAvg [42]. Across these settings, MAPFL consistently improves average accuracy and AUC under shared hyperparameters, with gains of about 1.1%–3.4% while also exhibiting faster and more stable convergence, suggesting that the framework is robust to modality changes and heterogeneous data partitions.

In summary, this paper makes the following contributions:

1. We formalise multi-shift heterogeneity in federated human-centric sensing, including label, concept, feature, and mixture shifts, and analyse its implications for robust optimization and client-level personalisation in FL.
2. We propose MAPFL, a robust personalised federated learning framework organised into three modules: a *Dual-Path Modular Backbone* (M1), a *Gradient-Aligned Mixture Controller* (M2), and a *Plateau-Scheduled Decoupling* mechanism (M3). These modules jointly realise a modality-agnostic and tuning-efficient design that (i) recovers global-only FL, layer-wise personalisation, and mixture-based PFL as special cases, and (ii) introduces a new encoder–head level, gradient-aligned control of sharing.
3. We conduct extensive experiments on federated visual sentiment classification and human activity recognition, using MobileNetV2 and convolutional HAR backbones. MAPFL consistently outperforms strong FL and PFL baselines under a unified configuration, suggesting that it provides a practical building block for future in-hospital and out-of-hospital human-centric sensing applications.

2 | Related Work

This section reviews relevant literature from three perspectives that underpin the design of MAPFL: (i) federated learning for generalisation, robustness, and fairness; (ii) personalised federated learning; and (iii) human-centric sensing in visual sentiment and human activity recognition.

2.1 | Federated Learning for Generalisation, Robustness and Fairness

Federated learning (FL) was originally proposed as a framework for training shared models over decentralised data without centralising raw samples [1]. The canonical FedAvg algorithm alternates between broadcasting a global model, performing several local stochastic gradient steps on each participating client, and averaging the resulting updates. While simple and communication-efficient, FedAvg and its early variants implicitly assume that client data are approximately independent and identically distributed, and can perform poorly when client distributions are highly heterogeneous [1, 8, 13].

To improve robustness under non-IID data, a range of optimization and aggregation strategies have been proposed. FedProx adds a proximal term to the local objective to mitigate client drift and stabilise convergence when local updates are large [13]. SCAFFOLD introduces control variates to correct for 'client drift' in local gradients, thereby reducing variance in the global update [14]. In the presence of feature distribution shift, FedBN keeps batch-normalisation statistics local while sharing the remaining parameters, leading to substantial gains on cross-domain medical imaging benchmarks [37]. Other methods, such as MOON [38], FedNTD [39], FedLC [40], FedBABU [41], FedDBE [26], and reinforcement-learning-based client selection schemes [43], further enrich the design space by incorporating contrastive learning, distillation, calibrated logits, representation fine-tuning, representation-space correction, and adaptive scheduling. These approaches primarily target *global generalisation* and *optimization robustness* in the presence of non-IID data and partial participation.

Beyond optimization stability, recent surveys emphasise that FL systems should be assessed along the dimensions of *generalisation*, *robustness*, and *fairness* [2, 8]. Ye et al. [2] discuss heterogeneous FL at scale, highlighting device and data heterogeneity as key obstacles to robustness and fairness. Huang et al. [8] argue that robust federated learning should consider generalisation, robustness, and fairness jointly rather than in isolation. These perspectives motivate a move from purely global models to more refined mechanisms that can adapt to client heterogeneity while maintaining global consistency. In this paper we take these perspectives as a motivation for our design, but our empirical study focuses primarily on global generalisation and client-level personalisation under heterogeneous clients; fairness-specific objectives and metrics are therefore left to future work.

Several works further explore system-level aspects of FL. Guo et al. [28] design a dynamic differential privacy mechanism for FL that adjusts noise injection according to training progress, providing privacy guarantees while preserving learning quality. Lyu et al. [29] study incentive mechanisms for FL participation, modelling privacy costs and rewards via reverse auctions and negotiations. Jiang and Hu [30] investigate decentralised FL with adaptive partial gradient aggregation, enabling clients to exchange partial updates without a central server. Hui et al. [31] propose Ada-FFL, which incorporates adaptive computing fairness considerations into FL. In the mobile and IoT context, Zhou et al. [32] propose device scheduling schemes for edge-assisted FL, jointly considering communication constraints and learning performance. These studies highlight that robust FL deployments must account for communication, privacy, incentives, and resource heterogeneity in addition to algorithmic design.

In domain-specific applications, FL has been used to integrate multi-centre medical data, where institutions have distinct distributions due to acquisition protocols and patient populations [4, 5]. Xu et al. [4] survey FL in healthcare informatics, emphasising privacy, regulatory compliance, and data heterogeneity. Liu et al. [5] show that foundation models can be adapted to multi-centre tuberculosis diagnosis via federated regularisation and model-contrastive learning, demonstrating the feasibility of combining pre-trained models with FL to improve cross-centre generalisation. In mobile communication networks, Kang et al. [6] analyse reliability issues and propose mechanisms to support reliable FL for mobile users. These works substantiate the importance of robust FL in safety-critical and human-centric domains.

The approach proposed in this paper builds on these developments but focuses specifically on *personalised* FL for human-centric sensing, under a multi-shift heterogeneity setting where both label and feature distributions vary across clients. Rather than introducing a new global optimiser, we seek a model-level mechanism that can adaptively share and personalise different components, while remaining compatible with the generalisation–robustness–fairness perspective articulated in the FL literature [2, 8, 12].

2.2 | Personalised Federated Learning

Personalised federated learning (PFL) aims to provide each client with a tailored model that better matches its local data distribution while still leveraging information from other clients [12]. This is particularly appealing for human-centric applications where user preferences, devices, and data collection conditions may differ substantially. PFL approaches can be broadly grouped into several categories.

2.2.1 | Multi-Task and Meta-Learning Based PFL

Multi-task FL methods explicitly formulate each client model as a task in a multi-task learning framework. Federated multi-task learning (FMTL) [17] jointly optimises a collection of client-specific models with a shared regularisation structure, allowing them to share statistical strength while retaining differences. Clustered FL methods, such as the clustered FL framework of Sattler et al. [25], group clients into clusters based on similarity and learn separate models per cluster. These methods can accommodate heterogeneous client populations but require designing appropriate task relatedness or clustering criteria.

Meta-learning based PFL methods, such as the model-agnostic approach of Fallah et al. [16], aim to learn an initialisation that can be efficiently adapted to each client via a small number of local gradient steps. Such methods provide theoretical guarantees on the adaptation error and are well-suited to settings where clients have limited data. However, their performance and stability can be sensitive to hyper-parameter choices and task similarity, and extending them to heterogeneous modalities such as images and time-series sensors requires careful engineering.

2.2.2 | *Regularisation and Mixture-Based PFL*

Regularisation-based PFL introduces explicit terms that tie each client model to a reference global model. Ditto [44] formulates a bi-level optimization problem where clients optimise personalised models subject to a regularisation term towards the global model, aiming to achieve fairness and robustness across clients. More generally, regularisation-based PFL can be seen as adding a personalisation prior to the global optimization process.

Mixture-based PFL methods explicitly maintain a *global* and a *personal* model on each client and learn an interpolation between them. Deng et al. [21] propose adaptive personalised FL (APFL), where each client optimises a mixing coefficient that interpolates between local and global models, with convergence guarantees under certain conditions. Building on this idea, FedALA [22] adaptively adjusts local aggregation weights based on client similarities, GPFL jointly learns generic and personalised feature information via separate feature extractors [23], and FedAS further investigates inconsistency in personalisation and global aggregation [45]. Recent work on class-wise federated averaging (cwFedAvg) considers class-level aggregation to better support efficient personalisation [42]. These methods are attractive due to their conceptual simplicity and the explicit control they provide over the global–local trade-off.

However, most mixture-based methods employ a *single* mixing parameter that scales the entire model or a small number of components uniformly. This coarse-grained control is less suitable when different layers or modules of the model experience different degrees of heterogeneity. For example, early convolutional layers in an image encoder may remain relatively transferable across clients, while classifier heads or high-level layers require stronger personalisation. In such cases, encoder–head or layer-wise personalisation could provide a more nuanced trade-off.

2.2.3 | *Partition-Based and Representation-Level PFL*

Partition-based PFL methods split the model into shared and private components. Arivazhagan et al. [18] propose FedPer, which shares early layers and keeps the final layers as personalisation layers on each client. In the context of HAR, Tu et al. [7] introduce FedDL, which dynamically shares different layers among clients according to their similarity. Prototype-based PFL, such as FedProto [24], operates in the representation space by sharing class-wise prototypes instead of full model parameters, allowing clients to personalise their classifiers while still benefiting from shared prototypes.

Representation-level approaches consider correcting or aligning features across domains. Zhang et al. [26] propose a representation-space method for eliminating domain bias in FL, showing that learning domain-invariant representations can significantly improve cross-client performance. Clustered FL can also be seen as learning multiple representations that serve different groups of clients [25]. FedBN [37] can similarly be interpreted as maintaining local batch statistics while sharing higher-level representations.

Partition-based methods align with the intuition that early layers capture generic patterns while later layers capture task- or user-specific information. However, they often require *manually* deciding which layers are shared and which are personal, and typically use a fixed split across all clients and tasks. This rigid design can be suboptimal when switching between visual tasks, where label subjectivity is prominent, and sensor tasks, where feature heterogeneity dominates.

2.2.4 | *Robustness, Heterogeneity and Personalised Fairness*

Several works explicitly consider robustness to data corruption or domain shift in FL. Fang et al. [46] study robust heterogeneous FL under data corruption and propose methods to improve performance when some clients provide noisy data. Huang et al. [47] analyse federated learning with domain shift from a prototype perspective, highlighting how prototype alignment can mitigate cross-domain differences. Ye et al. [2] survey heterogeneous FL, emphasising the interplay between system heterogeneity, data heterogeneity, and algorithm design. Tan et al. [12] provide a comprehensive overview of PFL methods and discuss how different classes of methods trade off personalisation, global performance, and fairness objectives. Together, these works suggest that robust PFL should not only enhance average performance but also avoid exacerbating disparities among clients with minority distributions or challenging data.

The MAPFL framework proposed in this paper can be viewed as a partition-based and mixture-based PFL method tailored for human-centric sensing, instantiated through three modules. Its *Dual-Path Modular Backbone* (M1) adopts encoder–head decoupling as in personalisation-layer and HAR-specific layer-sharing methods [7, 18], but maintains both global and personalised encoder–head copies on each client. The *Gradient-Aligned Mixture Controller* (M2) replaces static partitions with an adaptive, gradient-aligned interpolation mechanism inspired by mixture-based schemes [21], while the *Plateau-Scheduled Decoupling* module (M3) allows encoder and head to personalise to different extents once training stabilises. By focussing on encoder-level sharing and keeping heads local, MAPFL aims to balance global generalisation, robustness to multi-shift heterogeneity, and per-client personalisation with minimal manual tuning, distinguishing it from existing methods that either operate at the whole-model level or require fixed architecture choices.

2.3 | Human-Centric Sensing, Affective Computing and Activity Recognition

2.3.1 | *Visual Sentiment, Aesthetics and Affective Computing*

Visual sentiment analysis and aesthetic quality assessment have attracted extensive attention in the computer vision and multimedia communities. Early work by Machajdik and Hanbury [36] uses features inspired by psychology and art theory for affective image classification. More recent efforts build large-scale datasets and models for personalised image aesthetics assessment, such as PARA, which provides rich attributes and user-dependent aesthetic ratings [10]. These resources underline the inherently

subjective nature of visual sentiments and aesthetics, which is central to the multi-shift heterogeneity considered in this work.

Within the FL community, CAFL introduces conditional attention mechanisms for federated image emotion analysis, demonstrating that client-specific attention can improve performance under non-IID emotional labels [9]. In CAAI T-IT, studies on sentiment and emotion modelling, such as Bayesian estimation-based sentiment word embeddings [48] and self-training with maximum classifier discrepancy for EEG emotion recognition [49], further illustrate the importance of modelling fine-grained, user-related semantics in human-centric tasks. These studies, however, do not explicitly tackle personalised FL across heterogeneous clients.

In the context of visual backbones, MobileNetV2 has become a popular choice for resource-constrained devices, thanks to its inverted residuals and linear bottlenecks that significantly reduce computation and memory costs [34]. This makes it a natural candidate for image encoders in on-device human-centric sensing, as adopted in our MAPFL instantiation for visual sentiment tasks.

2.3.2 | *Human Activity Recognition and Federated Sensing*

Human activity recognition from smartphone and wearable sensors is a core building block in pervasive computing and mobile health. The UCI HAR dataset [11], which records accelerometer and gyroscope data from smartphones carried by volunteers performing common activities, has served as a benchmark for time-series classification and representation learning. However, this dataset and many follow-up works often assume centralised training, and do not fully capture the heterogeneity of real-world deployments.

Tu et al. [7] propose FedDL, a federated learning framework for HAR based on dynamic layer sharing, where different clients may share different subsets of layers depending on their similarity. This work shows that allowing layer-wise flexibility can improve performance in heterogeneous HAR scenarios. Other FL studies examine robust learning under data corruption [46] and domain shift [47], which are directly relevant to HAR where sensor noise, missing data, and domain shift are common. These contributions underscore the need for FL methods that can adapt to diverse sensor configurations and user behaviours while preserving overall robustness.

FL has also been applied to broader healthcare and human-centric applications, where physiological or clinical signals are distributed across institutions and devices [4, 5]. In such settings, data are often sensitive, and fairness across centres or demographic groups is critical. The methodologies developed for medical FL can inform the design of robust human-centric sensing systems beyond HAR, especially when foundation models and multi-modal data are involved [5].

2.3.3 | *Positioning of MAPFL*

From the above literature, two gaps emerge. First, most FL and PFL methods are evaluated either on vision benchmarks or on sensor benchmarks, but rarely on both under a unified configuration. This makes it difficult to assess whether a given method truly generalises across modalities and multi-shift heterogeneity. Second, while several works propose static or dynamic layer-sharing schemes [7, 18, 25], they often require task-specific decisions about which layers to share and which to personalise.

MAPFL addresses these gaps by providing a modality-agnostic, encoder–head based PFL framework organised into three modules that can be applied to both visual sentiment and HAR tasks with minimal architectural changes. The *Dual-Path Modular Backbone* (M1) defines the shared encoder–head parameterisation and dual-path optimization, the *Gradient-Aligned Mixture Controller* (M2) enables adaptive sharing of encoder and head parameters across clients, and the *Plateau-Scheduled Decoupling* module (M3) adjusts the granularity of personalisation over the course of training. Together, these modules keep classifier heads fully local to capture subjective label semantics while allowing encoders to interpolate between global and personalised behaviour. In this way, MAPFL sits at the intersection of robust optimization and client-level personalisation in FL [2, 8], while being tailored to the specific requirements of human-centric sensing. Conceptually, M1 is closest to layer- and head-personalisation methods such as FedPer and FedBABU, but it retains an explicit dual-path structure that keeps a fully global encoder–head copy on each client. M2 is related to mixture-based approaches such as APFL, FedALA, and GPFL, yet differs in that it attaches separate mixing coefficients to encoder and head and updates them via the gradient-aligned surrogate described in Section 3.3. M3 plays a complementary role to the above by providing a simple, loss-driven schedule that decides when encoder and head should start to diverge. Taken together, these modules position MAPFL not as a competing objective to existing PFL families, but as a unifying encoder–head framework that clarifies when and how global and personalised components should be combined. In the next section, we detail the MAPFL methodology and explain how its design choices are informed by the related work discussed above.

3 | Methodology

This section presents our Modular Adaptive Personalised Federated Learning (MAPFL) framework. We begin by formalising personalised FL under multi-type distribution shifts and introducing a modular encoder–head parameterisation. We then organise MAPFL into three interacting modules: (M1) a *Dual-Path Modular Backbone* that maintains global and personalised encoder–head copies on each client, (M2) a *Gradient-Aligned Mixture Controller* that learns module-level mixing coefficients, and (M3) a *Plateau-Scheduled Decoupling* mechanism that adjusts the granularity of personalisation over time. Finally, we describe round-end mixing and server aggregation, summarise the overall procedure in Algorithm 1, and relate MAPFL to existing global and personalised FL paradigms (Figure 1).

3.1 | Preliminaries and Sources of Modular Heterogeneity

We consider a standard cross-device FL setup with N clients indexed by $i \in \{1, \ldots, N\}$. Client i holds a local dataset

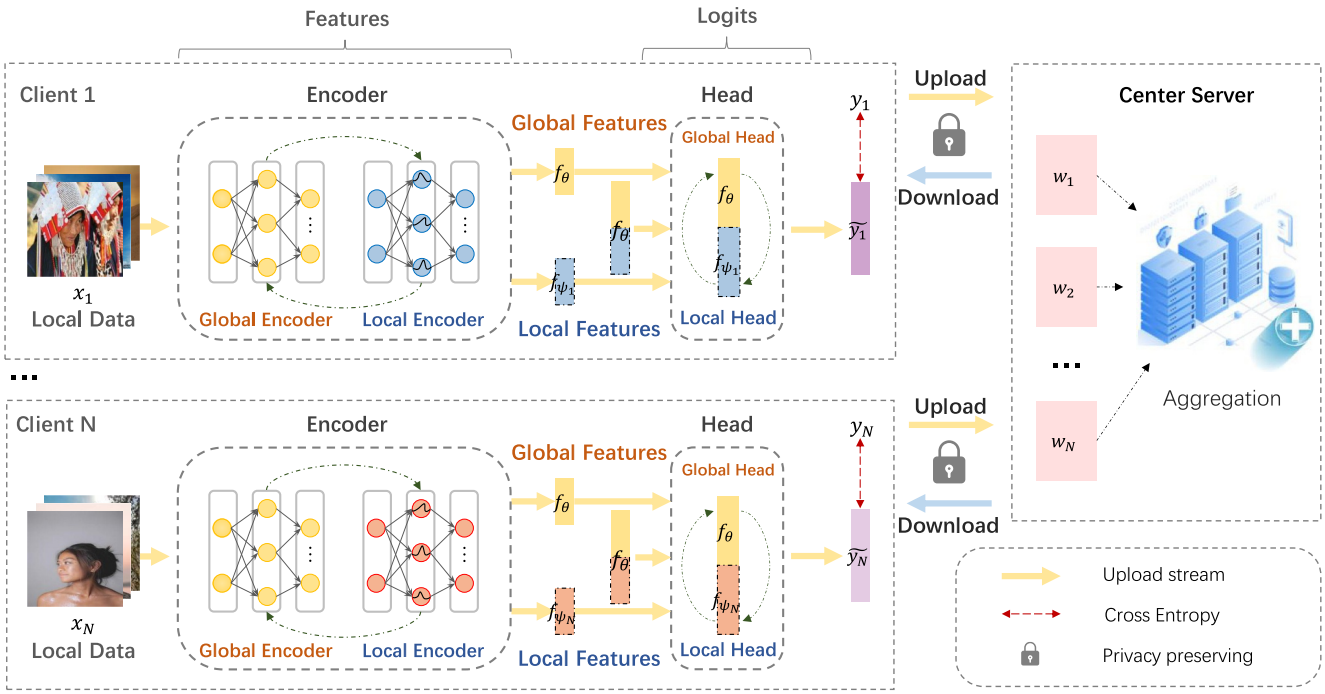


FIGURE 1 | Overall pipeline of the proposed MAPFL framework. At each communication round, the server broadcasts the current global encoder and head $(\theta_b^{(t)}, \theta_h^{(t)})$ to a subset of clients. Each participating client maintains two coupled models: (i) a *global copy* w_i that remains aggregatable across clients, and (ii) a *personalised model* v_i that stays on-device. Within each local epoch, both models are updated on every mini-batch through a *global path* (updating w_i) and a *personalised path* (updating v_i), while the modular mixing coefficients $(\alpha_{b,i}, \alpha_{h,i})$ are adapted using gradient-aligned rules. A client-side plateau detector monitors the personalised loss and, once improvement saturates, switches from a coupled regime $(\alpha_{b,i} \approx \alpha_{h,i})$ to a fully decoupled regime where encoder and head coefficients are optimised separately. At the end of the round, the personalised model is refreshed by mixing it with the updated global copy, whereas only the global copy is uploaded for weighted averaging on the server. In this way MAPFL learns client-specific encoder and head sharing patterns under heterogeneous combinations of feature, label, and concept shifts, without changing the standard communication pattern.

$S_i = \{(x_{ij}, y_{ij})\}_{j=1}^{n_i}$ of size n_i, drawn from an unknown distribution D_i. We denote client weights $p_i = n_i / \sum_{k=1}^{N} n_k$. The goal of personalised FL (PFL) is to learn client-specific models $\{h_i\}_{i=1}^{N}$ that leverage cross-client information while respecting client-level heterogeneity, without sharing raw data.

Human-centric sensing naturally exhibits multiple types of shift across clients: feature-level shift in $P_i(x)$ (e.g., different sensors or viewpoints), label-prior shift in $P_i(y)$ (e.g., different class frequencies), and concept shift in $P_i(y|x)$ (e.g., subjective emotion or aesthetic ratings). These shifts affect different parts of a deep network to different extents: encoders tend to learn reusable feature representations, while classifier heads must accommodate client-specific semantics and label priors. A single global model, or even a single scalar personalisation knob applied uniformly to all layers, is often too coarse to capture such modular heterogeneity.

To make this modular structure explicit, we decompose each model into a feature encoder and a classifier head. Let $f_{\theta_b}: \mathcal{X} \to \mathbb{R}^{d_b}$ and $g_{\theta_h}: \mathbb{R}^{d_b} \to \mathbb{R}^m$ denote an encoder and a head parameterised by $\theta_b \in \mathbb{R}^{d_b}$ and $\theta_h \in \mathbb{R}^{d_h}$, respectively. In classical global FL, all clients share a single parameter vector $\theta = (\theta_b, \theta_h)$ and use the global model

$$h(x; \theta) = g_{\theta_h}(f_{\theta_b}(x)). \tag{1}$$

The global model is trained by minimising the weighted empirical risk

$$F(\theta) = \sum_{i=1}^{N} p_i F_i(\theta), \quad F_i(\theta) = \frac{1}{n_i} \sum_{j=1}^{n_i} \ell\big(h(x_{ij}; \theta), y_{ij}\big), \tag{2}$$

where $\ell(\hat{y}, y)$ is a supervised loss such as cross-entropy.

In contrast, PFL aims at learning client-specific models h_i that share structure but can deviate from the global optimum whenever local distributions D_i differ significantly. A natural strategy, instantiated for example by APFL, is to maintain both a shared component and a personalised component on each client and to interpolate between them using a mixing coefficient. MAPFL adopts and extends this mixture view at the modular level: M1 defines global and personalised copies of both the encoder and the head, M2 learns mixing coefficients for each module and each client, and M3 schedules when these

coefficients should behave as a single shared weight or as fully decoupled encoder/head parameters.

3.2 | M1: Dual-Path Modular Backbone

The first module of MAPFL defines a modular mixture of global and local encoder–head parameters together with a dual-path training scheme on each client.

3.2.1 | *Modular Mixture of Global and Local Parameters*

For each client i, MAPFL keeps a pair of global parameters (θ_b, θ_h) and personalised parameters $(\psi_{b,i}, \psi_{h,i})$. Two scalar coefficients $\alpha_{b,i}, \alpha_{h,i} \in [0,1]$ control the interpolation between global and personalised parameters for the encoder (base) and head modules, respectively. The mixed modules are defined by

$$\vartheta_{b,i} = \alpha_{b,i}\,\psi_{b,i} + (1-\alpha_{b,i})\,\theta_b, \tag{3}$$

$$\vartheta_{h,i} = \alpha_{h,i}\,\psi_{h,i} + (1-\alpha_{h,i})\,\theta_h. \tag{4}$$

The personalised model of client i is then

$$h_i(x) = g_{\vartheta_{h,i}}\big(f_{\vartheta_{b,i}}(x)\big). \tag{5}$$

Different settings of $(\alpha_{b,i}, \alpha_{h,i})$ recover many existing behaviours:

1. $(\alpha_{b,i}, \alpha_{h,i}) = (0,0)$ yields a purely global model identical across clients (FedAvg-style).
2. $(\alpha_{b,i}, \alpha_{h,i}) = (1,1)$ yields purely local training without collaboration.
3. $(\alpha_{b,i}, \alpha_{h,i}) = (0,1)$ corresponds to a shared encoder with fully personalised heads, similar to head-personalisation methods.
4. $\alpha_{b,i} = \alpha_{h,i} = \alpha_i \in (0,1)$ reduces to a single-coefficient mixture between global and local models, analogous to APFL when the network is treated as a single block.

MAPFL generalises these cases by learning $(\alpha_{b,i}, \alpha_{h,i})$ from data for each client, allowing the degree of sharing to differ between representation and decision layers and to evolve during training.

3.2.1.1 | *Local Modular Objective and Federated Objective.* Given the mixed parameters $(\vartheta_{b,i}, \vartheta_{h,i})$, the empirical loss of client i is

$$L_i(\theta_b, \theta_h, \psi_{b,i}, \psi_{h,i}, \alpha_{b,i}, \alpha_{h,i}) = \frac{1}{n_i}\sum_{j=1}^{n_i} \ell\big(g_{\vartheta_{h,i}}(f_{\vartheta_{b,i}}(x_{ij})), y_{ij}\big) + r(\alpha_{b,i}, \alpha_{h,i}), \tag{6}$$

where r is a light-weight regulariser on the mixing coefficients that prevents pathological solutions and stabilises optimization. In practice we use a simple ℓ_2 penalty

$$r(\alpha_{b,i}, \alpha_{h,i}) = \lambda_\alpha(\alpha_{b,i}^2 + \alpha_{h,i}^2), \tag{7}$$

which gently discourages extreme values and plays the role of a consistency regulariser for the coefficients.

The federated objective is to minimise the weighted sum of local losses:

$$\min_{\Theta} F(\Theta) = \sum_{i=1}^{N} p_i\, L_i(\theta_b, \theta_h, \psi_{b,i}, \psi_{h,i}, \alpha_{b,i}, \alpha_{h,i}), \tag{8}$$

where Θ denotes the collection of all trainable variables. By optimising $\alpha_{b,i}$ and $\alpha_{h,i}$ jointly with global and local weights, the Dual-Path Modular Backbone lets each client decide how much it should rely on global versus local representations and decision boundaries at each module. Table 1 summarises the main notation used in MAPFL.

3.2.2 | *Dual-Path Local Optimization*

Directly optimising Equation (8) in a distributed setting is challenging because global parameters (θ_b, θ_h) and personalised parameters $(\psi_{b,i}, \psi_{h,i})$ are coupled through the mixtures $(\vartheta_{b,i}, \vartheta_{h,i})$ and the client-specific coefficients $(\alpha_{b,i}, \alpha_{h,i})$. The second component of M1 therefore maintains two coupled models on each selected client: a local copy of the global model and a fully personalised model, updated along two optimization paths.

At communication round t, the server broadcasts the current global parameters $(\theta_b^{(t)}, \theta_h^{(t)})$ to a subset $C^{(t)}$ of clients. On each selected client $i \in C^{(t)}$ we maintain:

1. A global copy $w_i = (\theta_{b,i}, \theta_{h,i})$, initialised as $(\theta_{b,i}, \theta_{h,i}) \leftarrow (\theta_b^{(t)}, \theta_h^{(t)})$;
2. A personalised model $v_i = (\psi_{b,i}, \psi_{h,i})$, which persists across rounds;
3. Modular personalisation coefficients $(\alpha_{b,i}, \alpha_{h,i})$.

During local training MAPFL alternates two optimization paths on each mini-batch $B_i \subset S_i$.

3.2.2.1 | *Global Path.* The global path updates the local copy w_i using the global parameters in both base and head. Given a mini-batch B_i, the global-path loss is

TABLE 1 | Summary of main notation used in MAPFL.

Symbol	Description
N	Number of clients
S_i	Local dataset on client i
n_i	Number of samples on client i
p_i	Client weight $n_i / \sum_k n_k$
x_{ij}, y_{ij}	j-th sample and label on client i
$f_{\theta_b}, g_{\theta_h}$	Encoder and head with global parameters
$f_{\psi_{b,i}}, g_{\psi_{h,i}}$	Encoder and head with personalised parameters
$\vartheta_{b,i}, \vartheta_{h,i}$	Mixed base and head parameters (Equations 3 and 4)
$\alpha_{b,i}, \alpha_{h,i}$	Modular personalisation coefficients for client i
λ_α	Coefficient regularisation weight in Equation (7)
$\Pi_{[0,1]}(\cdot)$	Projection (clipping) operator onto [0, 1]
$\overline{L}_i^{(t)}$	Round-averaged loss on client i at round t
$\ell(\cdot,\cdot)$	Per-sample supervised loss
T	Number of communication rounds
E	Number of local epochs per round
$\eta_w, \eta_v, \eta_\alpha$	Learning rates for w_i, v_i, and $(\alpha_{b,i}, \alpha_{h,i})$

$$L_i^g(w_i) = \frac{1}{|B_i|} \sum_{(x,y)\in B_i} \ell\big(g_{\theta_{h,i}}\big(f_{\theta_{b,i}}(x)\big), y\big), \tag{9}$$

and we perform a stochastic gradient step on w_i:

$$w_i \leftarrow w_i - \eta_w \nabla_{w_i} L_i^g(w_i). \tag{10}$$

This path ensures that the shared model continues to track a federation-wide representation and decision rule compatible with FedAvg-style aggregation.

3.2.2.2 | *Personalised Path.* The personalised path updates the client-specific model v_i using the same mini-batch:

$$L_i^p(v_i) = \frac{1}{|B_i|} \sum_{(x,y)\in B_i} \ell\big(g_{\psi_{h,i}}\big(f_{\psi_{b,i}}(x)\big), y\big), \tag{11}$$

followed by a gradient step:

$$v_i \leftarrow v_i - \eta_v \nabla_{v_i} L_i^p(v_i). \tag{12}$$

The two paths can be interpreted as maintaining a global trajectory that remains aggregatable across clients and a local trajectory that adapts aggressively to client-specific patterns. The mixtures in Equations (3) and (4), controlled by M2 and M3, combine these trajectories into the personalised predictor h_i used at inference time.

3.3 | M2: Gradient-Aligned Mixture Controller

A key feature of MAPFL is the adaptive update of $(\alpha_{b,i}, \alpha_{h,i})$, which determines the degree of sharing for the base and head on each client. To make the dependence on the mixed modules explicit, it is convenient to view the local objective as a function of $(\vartheta_{b,i}, \vartheta_{h,i})$:

$$L_i(\vartheta_{b,i}, \vartheta_{h,i}) = \frac{1}{n_i} \sum_{j=1}^{n_i} \ell\big(g_{\vartheta_{h,i}}\big(f_{\vartheta_{b,i}}(x_{ij})\big), y_{ij}\big) + r(\alpha_{b,i}, \alpha_{h,i}). \tag{13}$$

By the chain rule, the exact gradients of L_i with respect to $\alpha_{b,i}$ and $\alpha_{h,i}$ are

$$\frac{\partial L_i}{\partial \alpha_{b,i}} = \langle \nabla_{\vartheta_{b,i}} L_i(\vartheta_{b,i}, \vartheta_{h,i}),\ \psi_{b,i} - \theta_b \rangle, \tag{14}$$

$$\frac{\partial L_i}{\partial \alpha_{h,i}} = \langle \nabla_{\vartheta_{h,i}} L_i(\vartheta_{b,i}, \vartheta_{h,i}),\ \psi_{h,i} - \theta_h \rangle, \tag{15}$$

where $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product over parameters. Computing the mixed-module gradients $\nabla_{\vartheta_{b,i}} L_i$ and $\nabla_{\vartheta_{h,i}} L_i$ exactly would require additional backward passes and thus substantially increase local computation. To keep the controller lightweight, we approximate these gradients using the information already available from the two optimization paths. Intuitively, the mixed parameters $(\vartheta_{b,i}, \vartheta_{h,i})$ in Equation (13) lie on the line segment between the global and personalised encoder–head copies, so the exact derivatives $\partial L_i / \partial \alpha_{b,i}$ and $\partial L_i / \partial \alpha_{h,i}$ in Equations (14) and (15) measure how the local objective changes when we move along this segment. Under a first-order Taylor approximation of L_i around the current parameters, these directional derivatives can be estimated from suitable convex combinations of the gradients computed on the global and personalised paths. Concretely, the Gradient-Aligned Mixture Controller reuses the gradients already computed on the two paths. We define

$$\begin{aligned} g_i^{g,b} &= \nabla_{\theta_{b,i}} L_i^g(w_i), \quad g_i^{g,h} = \nabla_{\theta_{h,i}} L_i^g(w_i), \\ g_i^{p,b} &= \nabla_{\psi_{b,i}} L_i^p(v_i), \quad g_i^{p,h} = \nabla_{\psi_{h,i}} L_i^p(v_i), \end{aligned} \tag{16}$$

and approximate the mixed-module gradients by convex combinations of global and personalised gradients:

$$\tilde{g}_i^b = \alpha_{b,i} g_i^{p,b} + (1 - \alpha_{b,i}) g_i^{g,b}, \tag{17}$$

$$\tilde{g}_i^h = \alpha_{h,i} g_i^{p,h} + (1 - \alpha_{h,i}) g_i^{g,h}. \tag{18}$$

This yields a first-order surrogate of the true gradients with respect to $(\alpha_{b,i}, \alpha_{h,i})$; as long as the loss surface is locally smooth and the coefficients are updated with sufficiently small steps, moving against these surrogate gradients decreases the local objective on average, a behaviour that we also observe empirically in Section 4.

Substituting these approximations into Equations (14) and (15) and including the regulariser Equation (7) yields surrogate gradients

$$\hat{g}_i^{\alpha_b} = \langle \psi_{b,i} - \theta_b,\ \tilde{g}_i^b \rangle + \lambda_\alpha \alpha_{b,i}, \tag{19}$$

$$\hat{g}_i^{\alpha_h} = \langle \psi_{h,i} - \theta_h,\ \tilde{g}_i^h \rangle + \lambda_\alpha \alpha_{h,i}. \tag{20}$$

3.3.1 | *Intuition (Agreement, Disagreement, and Weak Correlation)*

The vector $\psi_{b,i} - \theta_b$ points from the global encoder to the personalised encoder of client i. The inner product $\langle \psi_{b,i} - \theta_b, \tilde{g}_i^b \rangle$ can be viewed as a (first-order) directional derivative of the local objective along this 'global → personalised' direction: if it is *positive*, moving $\alpha_{b,i}$ upward (placing more weight on $\psi_{b,i}$) tends to *increase* the loss, so the projected update in Equation (21) decreases $\alpha_{b,i}$ and relies more on the global encoder; if it is *negative*, moving towards $\psi_{b,i}$ tends to *decrease* the loss, and the update increases $\alpha_{b,i}$, strengthening personalisation.

In practice, the magnitude and sign of this signal are closely related to how the two optimization paths agree on the update direction: (i) when the global-path and personalised-path gradients are strongly *aligned* (high cosine similarity), the two paths suggest compatible improvements and the controller typically keeps $\alpha_{b,i}$ relatively small or changes it slowly (more sharing); (ii) when they are nearly *orthogonal* (weak correlation), the controller receives a weak/noisy signal and $\alpha_{b,i}$ changes little; (iii) when they are clearly *opposed*, the shared update conflicts with the client's descent direction and the controller increases $\alpha_{b,i}$ to protect client-specific adaptation. The same intuition applies to the head coefficient $\alpha_{h,i}$, enabling encoder and head to mix differently.

Given the surrogate gradients, a generic projected gradient step on the coefficients reads

$$\alpha_{b,i} \leftarrow \Pi_{[0,1]}\left(\alpha_{b,i} - \eta_\alpha \hat{g}_i^{\alpha_b}\right), \tag{21}$$

$$\alpha_{h,i} \leftarrow \Pi_{[0,1]}\left(\alpha_{h,i} - \eta_\alpha \hat{g}_i^{\alpha_h}\right), \tag{22}$$

where $\Pi_{[0,1]}$ denotes projection onto the unit interval [0, 1] (i.e., element-wise clipping). In MAPFL these updates are further structured by the Plateau-Scheduled Decoupling module described next.

3.4 | M3: Plateau-Scheduled Decoupling

While the generic updates Equations (21) and (22) allow the base and head to adapt independently, directly optimising two free coefficients per client from the beginning can be unstable, especially when the backbone is far from convergence. Empirically we observe that different modules often benefit from different personalisation regimes at different phases of training: early on, a single global–local trade-off suffices; later, encoder and head may require distinct levels of personalisation. M3 therefore adopts a two-phase schedule for $(\alpha_{b,i}, \alpha_{h,i})$ on each client.

3.4.1 | *Implementation Summary*

Before the plateau trigger, each client updates a *single shared* coefficient per mini-batch: we compute $\hat{g}_i^{\text{avg}}$ (Equation 25), apply Equations (26) and (27), and re-synchronize $\alpha_{b,i}$ and $\alpha_{h,i}$ via Equation (28). After the plateau trigger, the client switches to the decoupled phase and updates $\alpha_{b,i}$ and $\alpha_{h,i}$ independently per mini-batch using the reweighted steps in Equations (30) and (31).

3.4.2 | *Client-Side Plateau Detection*

Each client tracks the evolution of its personalised-path loss L_i^p. Let $\overline{L}_i^{(t)}$ denote the average of $L_i^p(v_i)$ over all local mini-batches in round t. The client maintains the best loss $\overline{L}_{i,\text{best}}$ seen so far and a counter c_i of consecutive rounds without significant improvement, given a tolerance $\delta_{\min} > 0$ and a patience parameter patience $\in \mathbb{N}$:

$$\overline{L}_{i,\text{best}} \leftarrow \min\left(\overline{L}_{i,\text{best}}, \overline{L}_i^{(t)}\right), \tag{23}$$

$$c_i \leftarrow \begin{cases} 0, & \overline{L}_i^{(t)} < \overline{L}_{i,\text{best}} - \delta_{\min}, \\ c_i + 1, & \text{otherwise.} \end{cases} \tag{24}$$

When $c_i \geq$ patience, client i declares that its personalised performance has plateaued and permanently switches to the decoupled phase described below.

3.4.3 | *Coupled Phase*

At the beginning of training, the base and head coefficients are constrained to move together: we maintain $\alpha_{b,i} \approx \alpha_{h,i}$ and update them using a shared scalar gradient. Formally, define the set of active modules $M_i \subseteq \{b, h\}$ (e.g., $M_i = \{b, h\}$ when both base and head are mixed) and form an averaged gradient

$$\hat{g}_i^{\text{avg}} = \frac{1}{|M_i|} \sum_{k \in M_i} \hat{g}_i^{\alpha_k}, \tag{25}$$

where α_b and α_h are treated symmetrically. We then update both coefficients with the same step and explicitly re-synchronize them:

$$\alpha_{b,i} \leftarrow \Pi_{[0,1]}\left(\alpha_{b,i} - \eta_\alpha \hat{g}_i^{\text{avg}}\right), \tag{26}$$

$$\alpha_{h,i} \leftarrow \Pi_{[0,1]}\left(\alpha_{h,i} - \eta_\alpha \hat{g}_i^{\text{avg}}\right), \tag{27}$$

$$\alpha_{b,i}, \alpha_{h,i} \leftarrow \frac{\alpha_{b,i} + \alpha_{h,i}}{2}. \tag{28}$$

In this phase, MAPFL behaves like a single-coefficient mixed model per client, closely related to mixture-based PFL such as APFL, but implemented with modular copies in preparation for later decoupling. In the coupled phase, Equation (28) is applied *after every mini-batch update* of the coefficients (i.e., once per local iteration inside the inner loop of Algorithm 1), so that the base and head share a single effective mixing weight early in training.

3.4.4 | *Decoupled (MAPFL) Phase*

Once plateau detection triggers for client i, the constraint is relaxed and $\alpha_{b,i}$ and $\alpha_{h,i}$ are updated independently. To reflect their relative importance, we reweight the effective step sizes according to the magnitudes of the surrogate gradients:

$$w_{b,i} = \frac{|\hat{g}_i^{\alpha_b}|}{|\hat{g}_i^{\alpha_b}| + |\hat{g}_i^{\alpha_h}|}, \quad w_{h,i} = \frac{|\hat{g}_i^{\alpha_h}|}{|\hat{g}_i^{\alpha_b}| + |\hat{g}_i^{\alpha_h}|}, \tag{29}$$

In practice, we compute $d = |\hat{g}_i^{\alpha_b}| + |\hat{g}_i^{\alpha_h}|$ and use a small constant ϵ (e.g., 10^{-12}) for numerical stability. If $d < \epsilon$, we set $(w_{b,i}, w_{h,i}) = (1/2, 1/2)$ and keep the coefficient update well-behaved; otherwise we use Equation (29). The decoupled updates then read

$$\alpha_{b,i} \leftarrow \Pi_{[0,1]}\left(\alpha_{b,i} - \eta_\alpha w_{b,i} \hat{g}_i^{\alpha_b}\right), \tag{30}$$

$$\alpha_{h,i} \leftarrow \Pi_{[0,1]}\left(\alpha_{h,i} - \eta_\alpha w_{h,i} \hat{g}_i^{\alpha_h}\right). \tag{31}$$

In implementation, the decoupled updates in Equations (30) and (31) are applied *once per mini-batch* inside the local inner loop of Algorithm 1; the plateau detector itself is evaluated once per round using the round-averaged loss $\overline{L}_i^{(t)}$.

Modules whose gradients carry stronger personalisation signals (larger $|\hat{g}_i^{\alpha_k}|$) receive proportionally larger updates. This two-phase design stabilises training in early rounds and, once decoupled, allows encoder and head to specialise differently on each client, capturing heterogeneous combinations of feature-level and label-level shifts.

3.5 | Round-End Mixing, Server Aggregation, and Algorithm Summary

After local training on client i in round t, MAPFL performs a modular mixing step that updates the personalised model v_i by blending it with the locally updated global copy w_i:

$$\psi_{b,i} \leftarrow \alpha_{b,i}\psi_{b,i} + (1 - \alpha_{b,i})\theta_{b,i}, \tag{32}$$

$$\psi_{h,i} \leftarrow \alpha_{h,i}\psi_{h,i} + (1 - \alpha_{h,i})\theta_{h,i}. \tag{33}$$

The personalised model $(\psi_{b,i}, \psi_{h,i})$ is then used for local evaluation and retained on the client for the next round, while only the updated global copy w_i and the local sample size n_i are sent to the server.

3.5.1 | *Reproducibility Notes*

3.5.1.1 | *Inference.* Although we define the mixed parameters $(\vartheta_{b,i}, \vartheta_{h,i})$ in Equations (3) and (4), in implementation we do *not* re-compute the mixture dynamically at every forward pass. Instead, the round-end mixing in Equations (32) and (33) updates $(\psi_{b,i}, \psi_{h,i})$, so the stored personalised model v_i already corresponds to the mixed model used for client-side evaluation and test-time inference.

3.5.1.2 | *Communication.* During training, clients transmit *only* the updated global copy $w_i = (\theta_{b,i}, \theta_{h,i})$ (i.e., encoder + head trainable weights) and the local sample size n_i to the server. Neither the mixing coefficients $(\alpha_{b,i}, \alpha_{h,i})$ nor any part of the personalised model $v_i = (\psi_{b,i}, \psi_{h,i})$ is transmitted.

3.5.1.3 | *Batch Normalisation (BN).* For backbones containing BN layers (e.g., MobileNetV2), each model copy (w_i and v_i) maintains its own BN running statistics. Following common FL practice under feature shift, BN running means/variances are kept local (not averaged on the server); only trainable weights are aggregated.

On the server, the global parameters are updated by standard weighted averaging:

$$\theta_b^{(t+1)} = \sum_{i \in C^{(t)}} \tilde{p}_i \, \theta_{b,i}, \qquad \theta_h^{(t+1)} = \sum_{i \in C^{(t)}} \tilde{p}_i \, \theta_{h,i}, \tag{34}$$

where $\tilde{p}_i = n_i / \sum_{k \in C^{(t)}} n_k$ are normalised weights over participating clients. The modular mixing thus happens entirely on the client side; communication cost remains the same as standard FedAvg because each client only transmits a single copy of the global model per round. On the other hand, maintaining both global and personalised copies and updating them along two optimization paths increases the amount of local computation per mini-batch compared to single-path baselines. In our experiments this overhead is acceptable for the considered visual and HAR backbones, but a more detailed profiling of latency and energy consumption on resource-constrained devices is an interesting direction for future work.

For analysis, we record the trajectories of $\alpha_{b,i}$ and $\alpha_{h,i}$ on the client side (in our simulator we can collect these logs after training). If desired in a real deployment, clients may *optionally* report these two scalars per round as lightweight monitoring metadata; this is not required by the MAPFL training protocol. Such metadata adds negligible bandwidth compared to model updates (e.g., two float32 values per round), but it may still reveal coarse information about a client's sharing preference, so we treat it as optional. As shown later in the experiments, these trajectories provide an interpretable view of how much each client relies on global knowledge at each module: clients with strong concept shift tend to converge to highly personalised heads, while those with milder heterogeneity rely more on the global encoder.

Algorithm 1 summarises the complete MAPFL training procedure. Conceptually, MAPFL can be viewed as a modular generalisation of mixture-based PFL: if we collapse encoder and head into a single block and enforce $\alpha_{b,i} = \alpha_{h,i}$ throughout training, MAPFL reduces to a single-coefficient mixture between global and local models, akin to APFL. The Dual-Path Modular Backbone (M1), Gradient-Aligned Mixture Controller (M2), and Plateau-Scheduled Decoupling (M3), however, enable MAPFL to capture richer patterns of heterogeneity and to expose interpretable per-module sharing profiles.

ALGORITHM 1 | MAPFL: Modular Adaptive Personalised Federated Learning.

Input: Number of communication rounds T; local epochs E; step sizes $\eta_w, \eta_v, \eta_\alpha$; regularization coefficient λ_α; plateau hyper-parameters (patience, $\delta_{\min}$).

Output: Trained global model $\left(\theta_b^{(T)}, \theta_h^{(T)}\right)$ and personalized models $\{v_i\}_i$.

Initialize global encoder and head $\left(\theta_b^{(0)}, \theta_h^{(0)}\right)$.

for $t = 0, \ldots, T-1$ **do**

  Server samples a subset of clients $C^{(t)}$ and broadcasts $\left(\theta_b^{(t)}, \theta_h^{(t)}\right)$.

  for *each client* $i \in C^{(t)}$ *in parallel* **do**

    Initialize local copy $w_i = (\theta_{b,i}, \theta_{h,i}) \leftarrow \left(\theta_b^{(t)}, \theta_h^{(t)}\right)$.

    if *client* i *participates for the first time* **then**

      Initialize personalized model $v_i = (\psi_{b,i}, \psi_{h,i}) \leftarrow \left(\theta_b^{(t)}, \theta_h^{(t)}\right)$.

      Initialize coefficients $\alpha_{b,i}, \alpha_{h,i} \in [0,1]$ (e.g. $\alpha_{b,i} = \alpha_{h,i} = 0.5$).

      Set $\bar{L}_{i,\text{best}} \leftarrow +\infty$, $c_i \leftarrow 0$, and decoupled$_i \leftarrow$ false.

    Reset the list that stores personalized-path losses.

    for $e = 1, \ldots, E$ **do**

      for *each mini-batch* $B_i \subset S_i$ **do**

        M1/global path: compute $L_i^g(w_i)$ using (9) and update w_i as in (10).

        M1/personalized path: compute $L_i^p(v_i)$ using (11), update v_i as in (12), and append $L_i^p(v_i)$ to the loss list.

        M2: compute surrogate gradients $\hat{g}_i^{\alpha_b}, \hat{g}_i^{\alpha_h}$ via Eqs. (16)–(20).

        if *decoupled*$_i$ *is* ***false*** **then**

          Form shared gradient $\hat{g}_i^{\text{avg}}$ as in (25).

          Update $(\alpha_{b,i}, \alpha_{h,i})$ using projected steps (26)–(27) and re-synchronize via (28). `// M3 coupled phase`

        else

          Compute weights $(w_{b,i}, w_{h,i})$ as in (29).

          Update $(\alpha_{b,i}, \alpha_{h,i})$ using reweighted projected steps (30)–(31). `// M3 decoupled phase`

    Compute average personalized loss $\bar{L}_i^{(t)}$ over stored L_i^p values.

    Update $(\bar{L}_{i,\text{best}}, c_i)$ via Eqs. (23)–(24).

    if $c_i \geq$ *patience* **then**

      Set decoupled$_i \leftarrow$ true.

    Round-end mixing (M1+M2): update personalized parameters $(\psi_{b,i}, \psi_{h,i})$ using Eqs. (32)–(33).

    Send updated global copy w_i and sample count n_i to the server.

  Server aggregation: update $\left(\theta_b^{(t+1)}, \theta_h^{(t+1)}\right)$ using weighted averaging (34).

4 | Experiments and Discussion

Our experiments are designed to answer three research questions (RQ1–RQ3):

RQ1. *Can MAPFL provide robust performance across different human-centric sensing tasks and dominant distribution shifts (concept/label shift* vs. *feature/mixture shift)?*

RQ2. *How does MAPFL behave in a label-scarce regime, especially on HAR where each client only has 10% labelled windows for training?*

RQ3. *Are the learned sharing patterns interpretable, and do the gains come from MAPFL's modular design rather than fragile hyper-parameter tuning?*

4.1 | Linking RQs to Experiments and Conclusions

To make the linkage explicit, we map each RQ to the corresponding experimental evidence and summarise the key takeaways:

1. *RQ1* (Sections 4.2 and 4.3; Tables 2–4; Figures 2–6). *Conclusion:* MAPFL consistently improves AUC/ACC on both ISC (dominated by label/concept shift) and HAR (dominated by feature/mixture shift), indicating robust cross-modality effectiveness under different heterogeneity types.
2. *RQ2* (Section 4.3; Table 4; Figure 6). *Conclusion:* Under 10% labelled data per client, MAPFL remains stable and achieves the best overall metrics, suggesting that encoder-level adaptive sharing is particularly beneficial when supervision is scarce.
3. *RQ3* (Sections 4.4 and 4.5; Figures 7 and 8). *Conclusion:* The learned mixing coefficients exhibit meaningful client-wise and module-wise patterns, and ablations indicate that MAPFL's improvements are not dependent on a narrow hyper-parameter choice.

We evaluate MAPFL on two representative benchmarks: an image-based sentiment classification benchmark (ISC) and a human activity recognition benchmark (HAR). On both datasets we compare against a broad set of global, regularisation-based and personalised FL baselines, and then analyse MAPFL's learned personalisation coefficients, ablations and hyper-parameters.

4.2 | Experimental Setup

4.2.1 | *Datasets and Tasks*

4.2.1.1 | *Visual Sentiment Analysis Benchmark (ISC).* For visual sentiment analysis, we utilise the image-based sentiment classification (ISC) benchmark, which aggregates three in-the-wild datasets: ARTphoto, ABSTRACT, and PARA. Each client corresponds to a user with their own subjective annotation style and local image collection, leading primarily to label and concept shifts. We consider two tasks:

1. *ISC-2 (2-class).* A binary sentiment classification task (configured as *ISC-2*) where images are labelled as *positive* or *negative.* This is a coarse-grained but still subjective task, with long-tailed label distributions across clients.
2. *ISC-8 (8-class).* A fine-grained 8-class emotion recognition task (configured as *ISC-8*) built on the same images but with richer categorical labels (e.g., joy, anger, sadness). Here subjectivity is stronger and local class distributions are highly imbalanced and skewed, making it a challenging testbed for personalised FL.

We adopt the same client partitioning as prior work: 24 clients participate in training, and each method is evaluated in both *pre-trained* and *from-scratch* settings with a MobileNetV2 backbone.

4.2.1.2 | *Summary of Heterogeneity.* Overall, ISC is dominated by subjective label/concept shift across users, while HAR is dominated by sensor-driven feature/mixture shift; the 10%

TABLE 2 | Performance comparison on the *ISC-2 task* (2 classes).

	MobileNetV2 (pre-trained)		MobileNetV2 (scratch)		Avg	
Method	**AUC (%)**	**ACC (%)**	**AUC (%)**	**ACC (%)**	**AUC (%)**	**ACC (%)**
APFL [21]	$84.11_{\pm 0.10}$	$77.84_{\pm 0.74}$	$81.50_{\pm 0.46}$	$75.39_{\pm 0.36}$	$82.81_{\pm 0.28}$	$76.62_{\pm 0.55}$
APPLE [50]	$82.62_{\pm 0.05}$	$75.81_{\pm 0.35}$	$78.63_{\pm 0.58}$	$74.79_{\pm 0.00}$	$80.62_{\pm 0.32}$	$75.30_{\pm 0.17}$
FedAvg [1]	$84.13_{\pm 0.20}$	$78.21_{\pm 0.86}$	$81.85_{\pm 0.88}$	$76.13_{\pm 0.96}$	$82.99_{\pm 0.54}$	$77.17_{\pm 0.91}$
FedBABU [41]	$87.89_{\pm 0.43}$	$\underline{81.90}_{\pm \underline{0.26}}$	$\underline{83.23}_{\pm \underline{2.04}}$	$\underline{78.86}_{\pm \underline{1.31}}$	$\underline{85.56}_{\pm \underline{1.24}}$	$\underline{80.38}_{\pm \underline{0.79}}$
FedBN [37]	$84.11_{\pm 0.20}$	$78.21_{\pm 0.75}$	$81.83_{\pm 0.46}$	$75.81_{\pm 1.01}$	$82.97_{\pm 0.33}$	$77.01_{\pm 0.88}$
FedDBE [26]	$84.63_{\pm 0.16}$	$78.44_{\pm 0.36}$	$82.87_{\pm 0.36}$	$77.79_{\pm 0.52}$	$83.75_{\pm 0.26}$	$78.12_{\pm 0.44}$
FedLC [40]	$84.10_{\pm 0.19}$	$77.84_{\pm 0.23}$	$81.29_{\pm 0.20}$	$74.98_{\pm 0.58}$	$82.69_{\pm 0.20}$	$76.41_{\pm 0.40}$
FedNTD [39]	$84.27_{\pm 0.26}$	$78.07_{\pm 0.53}$	$79.69_{\pm 0.61}$	$74.79_{\pm 0.00}$	$81.98_{\pm 0.43}$	$76.43_{\pm 0.27}$
FedProx [13]	$84.12_{\pm 0.21}$	$78.16_{\pm 0.82}$	$81.81_{\pm 0.53}$	$76.92_{\pm 0.96}$	$82.97_{\pm 0.37}$	$77.54_{\pm 0.89}$
MOON [38]	$84.13_{\pm 0.13}$	$78.53_{\pm 0.59}$	$81.53_{\pm 0.72}$	$76.08_{\pm 1.43}$	$82.83_{\pm 0.42}$	$77.31_{\pm 1.01}$
SCAFFOLD [14]	$83.39_{\pm 0.17}$	$76.92_{\pm 0.28}$	$79.28_{\pm 0.65}$	$74.79_{\pm 0.00}$	$81.34_{\pm 0.41}$	$75.86_{\pm 0.14}$
cwFedAvg [42]	$83.90_{\pm 0.19}$	$77.70_{\pm 0.30}$	$81.18_{\pm 0.57}$	$75.81_{\pm 0.79}$	$82.54_{\pm 0.38}$	$76.75_{\pm 0.55}$
MAPFL (ours)	$\mathbf{88.21}_{\pm \mathbf{0.07}}$	$\mathbf{82.46}_{\pm \mathbf{0.83}}$	$\mathbf{85.17}_{\pm \mathbf{0.41}}$	$\mathbf{79.96}_{\pm \mathbf{0.47}}$	$\mathbf{86.69}_{\pm \mathbf{0.24}}$	$\mathbf{81.21}_{\pm \mathbf{0.65}}$
Improve.	↑ 0.32	↑ 0.56	↑ 1.94	↑ 1.10	↑ 1.13	↑ 0.83

Note: Results are reported as mean ± std over $n = 3$ runs. Best in bold, second in underline. The red values represent the points of improvement of the proposed method relative to the strongest baseline method.

TABLE 3 | Performance comparison on the *ISC-8 task* (8 classes).

	MobileNetV2 (pre-trained)		MobileNetV2 (scratch)		Avg	
Method	**AUC (%)**	**ACC (%)**	**AUC (%)**	**ACC (%)**	**AUC (%)**	**ACC (%)**
APFL [21]	$74.74_{\pm 0.08}$	$39.80_{\pm 0.47}$	$72.17_{\pm 0.54}$	$38.64_{\pm 0.00}$	$73.45_{\pm 0.31}$	$39.22_{\pm 0.23}$
APPLE [50]	$73.06_{\pm 0.21}$	$39.57_{\pm 0.35}$	$72.70_{\pm 0.31}$	$38.64_{\pm 0.00}$	$72.88_{\pm 0.26}$	$39.11_{\pm 0.17}$
FedAvg [1]	$74.74_{\pm 0.12}$	$39.70_{\pm 0.17}$	$71.92_{\pm 0.48}$	$38.64_{\pm 0.00}$	$73.33_{\pm 0.30}$	$39.17_{\pm 0.09}$
FedBABU [41]	$\underline{79.92}_{\pm \underline{0.41}}$	$\underline{50.60}_{\pm \underline{0.92}}$	$\underline{75.37}_{\pm \underline{1.11}}$	$\underline{45.71}_{\pm \underline{0.23}}$	$\underline{77.65}_{\pm \underline{0.76}}$	$\underline{48.16}_{\pm \underline{0.58}}$
FedBN [37]	$74.74_{\pm 0.11}$	$39.61_{\pm 0.30}$	$71.86_{\pm 0.69}$	$38.64_{\pm 0.00}$	$73.30_{\pm 0.40}$	$39.12_{\pm 0.15}$
FedDBE [26]	$75.16_{\pm 0.07}$	$40.49_{\pm 0.07}$	$72.02_{\pm 0.57}$	$38.64_{\pm 0.00}$	$73.59_{\pm 0.32}$	$39.56_{\pm 0.04}$
FedLC [40]	$74.48_{\pm 0.08}$	$39.52_{\pm 0.68}$	$71.89_{\pm 0.78}$	$38.64_{\pm 0.00}$	$73.19_{\pm 0.43}$	$39.08_{\pm 0.34}$
FedNTD [39]	$75.12_{\pm 0.51}$	$39.84_{\pm 0.35}$	$72.98_{\pm 0.44}$	$38.64_{\pm 0.00}$	$74.05_{\pm 0.47}$	$39.24_{\pm 0.17}$
FedProx [13]	$74.74_{\pm 0.12}$	$39.61_{\pm 0.30}$	$71.86_{\pm 0.53}$	$38.64_{\pm 0.00}$	$73.30_{\pm 0.33}$	$39.12_{\pm 0.15}$
MOON [38]	$74.73_{\pm 0.12}$	$39.66_{\pm 0.07}$	$72.03_{\pm 0.76}$	$38.64_{\pm 0.00}$	$73.38_{\pm 0.44}$	$39.15_{\pm 0.04}$
SCAFFOLD [14]	$73.57_{\pm 0.28}$	$39.80_{\pm 0.13}$	$72.22_{\pm 0.43}$	$38.64_{\pm 0.00}$	$72.89_{\pm 0.35}$	$39.22_{\pm 0.07}$
cwFedAvg [42]	$74.91_{\pm 0.10}$	$39.84_{\pm 0.24}$	$71.85_{\pm 0.49}$	$38.64_{\pm 0.00}$	$73.38_{\pm 0.29}$	$39.24_{\pm 0.12}$
MAPFL (ours)	$\mathbf{83.23}_{\pm \mathbf{0.33}}$	$\mathbf{53.46}_{\pm \mathbf{0.41}}$	$\mathbf{77.63}_{\pm \mathbf{0.37}}$	$\mathbf{46.54}_{\pm \mathbf{0.39}}$	$\mathbf{80.43}_{\pm \mathbf{0.35}}$	$\mathbf{50.00}_{\pm \mathbf{0.40}}$
Improve.	↑ 3.31	↑ 2.86	↑ 2.26	↑ 0.83	↑ 2.78	↑ 1.84

Note: Results are reported as mean ± std over $n = 3$ runs. Best in bold, second in underline. The red values represent the points of improvement of the proposed method relative to the strongest baseline method.

TABLE 4 | Performance comparison on the *HAR benchmark* (6 classes).

	HARCNN (scratch, 10% train/90% test)	
Method	**AUC (%)**	**ACC (%)**
APFL [21]	$93.57_{\pm 0.25}$	$\underline{79.34}_{\pm \underline{0.55}}$
APPLE [50]	$92.86_{\pm 0.12}$	$79.27_{\pm 0.99}$
FedAvg [1]	$93.54_{\pm 0.12}$	$75.98_{\pm 0.21}$
FedBABU [41]	$74.65_{\pm 4.98}$	$41.89_{\pm 4.86}$
FedBN [37]	$93.54_{\pm 0.12}$	$75.99_{\pm 0.20}$
FedDBE [26]	$\underline{93.60}_{\pm \underline{0.20}}$	$79.32_{\pm 0.54}$
FedLC [40]	$93.59_{\pm 0.10}$	$75.83_{\pm 0.36}$
FedNTD [39]	$93.45_{\pm 0.13}$	$75.76_{\pm 0.12}$
FedProx [13]	$93.54_{\pm 0.12}$	$75.99_{\pm 0.20}$
MOON [38]	$93.54_{\pm 0.12}$	$75.99_{\pm 0.20}$
SCAFFOLD [14]	$93.52_{\pm 0.14}$	$74.05_{\pm 0.95}$
cwFedAvg [42]	$93.54_{\pm 0.12}$	$75.99_{\pm 0.20}$
MAPFL (ours)	$\mathbf{94.33}_{\pm \mathbf{0.19}}$	$\mathbf{81.77}_{\pm \mathbf{0.34}}$
Improve.	↑ 0.73	↑ 2.43

Note: Results are reported as mean ± std over $n = 3$ runs. Best in bold, second in underline. The red values represent the points of improvement of the proposed method relative to the strongest baseline method.

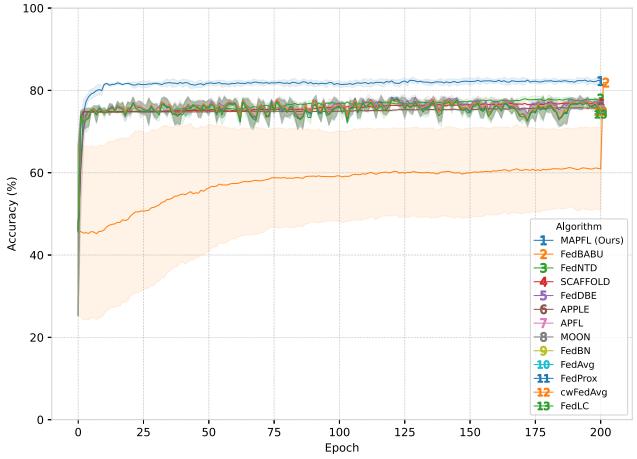


FIGURE 2 | Validation accuracy curves on the affective-computing benchmark (*ISC-2*) with a pre-trained MobileNetV2 backbone. MAPFL converges fastest and attains the highest validation accuracy among all methods, while strong personalised baselines such as APFL, APPLE and FedBABU plateau at lower levels.

labelled setting on HAR further models a realistic label-scarce wearable FL scenario.

4.2.1.3 | *Human Activity Recognition Benchmark (HAR).* For HAR, we use the specific *HAR* split configuration derived from the UCI HAR dataset [11]. Each client corresponds to a distinct user (subject) equipped with wearable sensors. The task is 6-class activity recognition (walking, walking upstairs, walking downstairs, sitting, standing, lying). In contrast to ISC, label semantics are largely objective and shared across users, while feature distributions differ significantly due to sensor placement, gait style and device noise, leading primarily to feature and mixture shifts.

To explicitly stress the label-scarce regime typical in mobile sensing, we use only 10% of the windows of each client for training and reserve the remaining 90% for testing. We choose 10% as a challenging yet realistic *cold-start* fraction: in wearable/sensor deployments, obtaining ground-truth activity labels often requires short supervised calibration sessions or user self-report, so only a small portion of on-device streams can be reliably annotated. We focus on the purely supervised FL setting and therefore treat the remaining 90% windows as strictly held-out

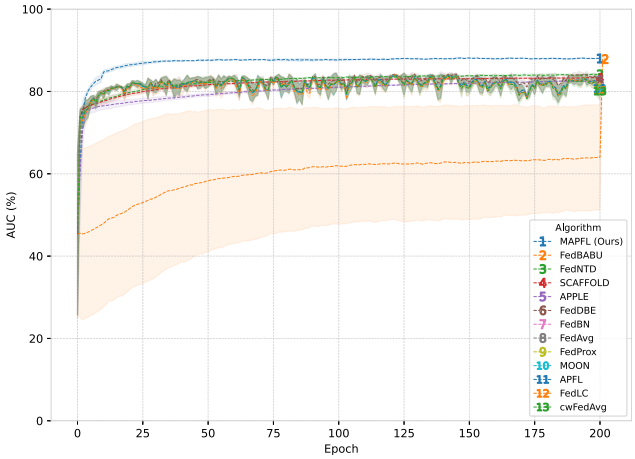


FIGURE 3 | Validation AUC curves on the affective-computing benchmark (*ISC-2*) with a pre-trained MobileNetV2 backbone. MAPFL consistently attains the highest AUC throughout training, showing that encoder-level personalisation improves both convergence speed and final generalisation under concept shift.

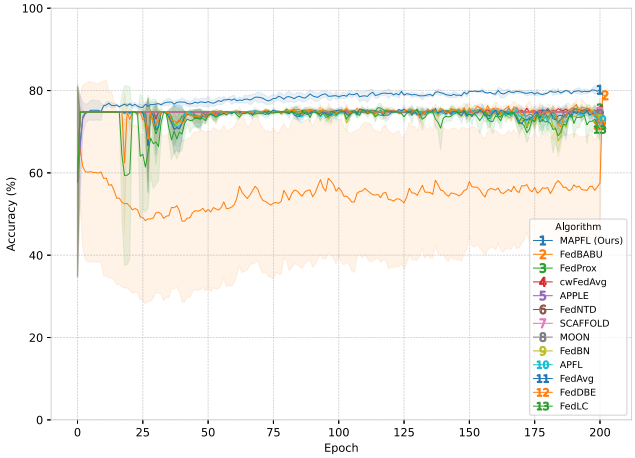


FIGURE 4 | Validation accuracy curves on the affective-computing benchmark (*ISC-2*) when MobileNetV2 is trained from scratch. Even without pre-training, MAPFL converges rapidly and stays clearly ahead of all baselines, while several methods stagnate at much lower accuracy or collapse to trivial solutions.

test data (not used for training or model selection). This setup highlights how well a method can transfer global knowledge and personalise from limited labelled data under strong subject-specific feature shift.

4.2.2 | *Baselines and Implementation Details*

We compare MAPFL with a diverse set of FL algorithms, covering global, regularisation-based, personalised and mixture-style methods: FedAvg, FedProx, FedLC, FedNTD, SCAFFOLD, cwFedAvg, FedBN, MOON, APFL, APPLE, FedBABU and FedDBE. They jointly cover representative strategies such as proximal regularisation, control variates, batch-normalisation based personalisation, contrastive representation learning, head-only personalisation and partition-level clustering.

4.2.2.1 | *Backbones.* On ISC we use MobileNetV2 as the backbone encoder. We evaluate both (i) an ImageNet pre-trained encoder with a randomly initialised classifier head, and (ii) a fully random initialisation where both encoder and head are trained from scratch. On HAR we use the HARCNN backbone implemented in the same codebase, following standard hyper-parameters for convolutional HAR models.

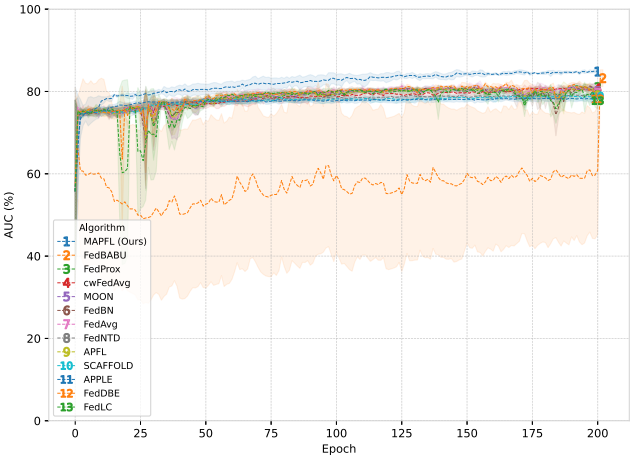


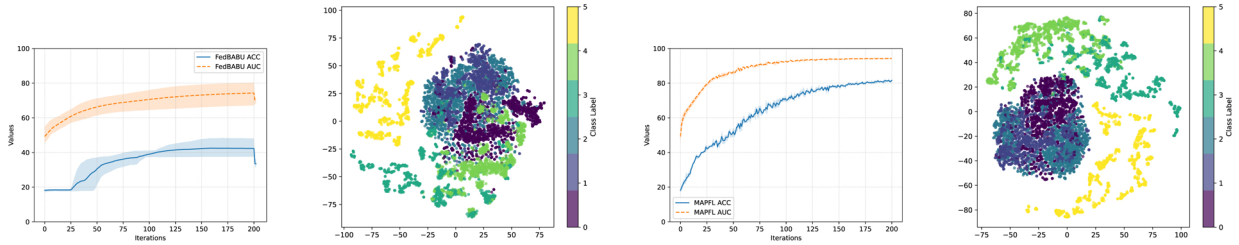
FIGURE 5 | Validation AUC curves on the affective-computing benchmark (*ISC-2*) with MobileNetV2 trained from scratch. MAPFL maintains a pronounced AUC advantage over both global and personalised baselines, indicating that it can build a strong shared representation from random initialisation under severe label and concept heterogeneity.

4.2.2.2 | *Training Recipe and FL Hyper-Parameters.* On both benchmarks we run 200 communication rounds. Each local update uses mini-batch SGD with a batch size of 16 and one local epoch per round. The client sampling ratio is set to 0.5 on ISC (12 of 24 clients are sampled per round) and 1.0 on HAR (all 30 clients participate at each round). All methods share the same optimiser, learning-rate schedule and weight decay on a given dataset:

1. On ISC we use a local learning rate of 10^{-3} with momentum 0.9 and weight decay 10^{-4}. The learning rate decays exponentially with factor 0.99 per communication round.
2. On HAR we follow common HAR settings and use a local learning rate of 10^{-1} with momentum 0.9 and weight decay 10^{-4}, again with exponential decay factor 0.99 per round.

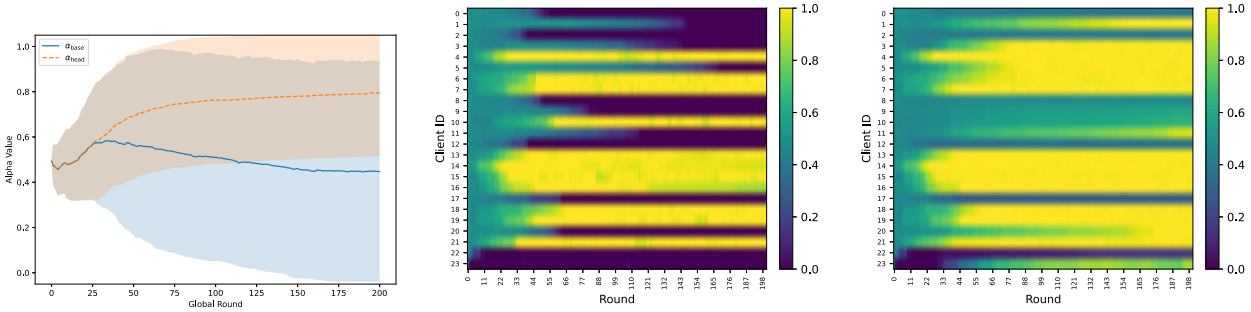
These choices are tuned once on a validation split and then kept *fixed across all algorithms* on each benchmark. This emphasises algorithmic robustness and fairness rather than per-method hyper-parameter tuning.

4.2.2.3 | *MAPFL-Specific Settings.* Unless otherwise stated, MAPFL initialises both encoder and head mixing coefficients to moderate values, $\alpha_{b,i}^{(0)} = \alpha_{h,i}^{(0)} = 0.5$ for all clients. The mixing coefficients are updated by the gradient-aligned rules with learning rate $\eta_\alpha = 0.1$, together with an ℓ_2 regulariser on the coefficients using $\lambda_\alpha = 10^{-3}$ in Equation (7). A client-side plateau detector monitors the personalised-path loss with patience = 5 and minimum improvement threshold $\delta_{\min} = 10^{-3}$.



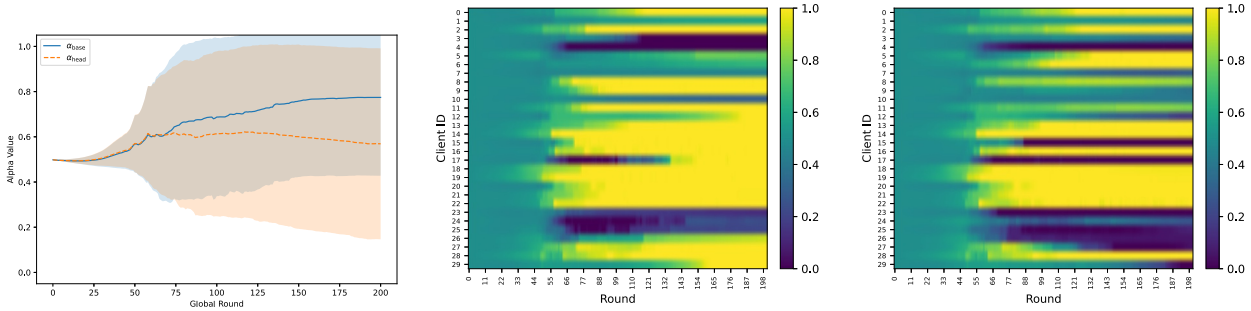
(a) (b) (c) (d)

FIGURE 6 | Comparison between FedBABU and MAPFL on the HAR benchmark with 10% train/90% test per client. (a) Validation accuracy and AUC of FedBABU over communication rounds, showing unstable and low performance. (b) t-SNE of FedBABU features, where activity clusters are heavily entangled. (c) Validation accuracy and AUC of MAPFL, which converge smoothly to substantially higher values. (d) t-SNE of MAPFL features, where the six activities form more compact and separated clusters. The contrast highlights that frozen-head personalisation in FedBABU is brittle under feature-level heterogeneity, whereas MAPFL learns a robust shared encoder.



(a) (b) (c)

FIGURE 7 | Evolution of personalisation coefficients on ISC-2. (a) Mean and variance of encoder and head coefficients (α_b, α_h) across clients as training proceeds. (b, c) Per-client heatmaps of α_b and α_h. MAPFL quickly moves away from the neutral initialisation and converges to distinct operating points for encoder and head, with relatively homogeneous encoder sharing but more diverse head coefficients, reflecting that visual features can be shared while sentiment decision boundaries are user-specific.



(a) (b) (c)

FIGURE 8 | Evolution of personalisation coefficients on HAR. (a) Mean and variance of (α_b, α_h) across clients. (b, c) Per-client heatmaps of encoder and head coefficients. Compared with the visual sentiment task, encoder coefficients here show larger client-to-client variation and tend to favour more local encoders, whereas head coefficients remain concentrated around intermediate values. This reflects that sensor characteristics are client-specific while activity semantics are largely shared, and provides an interpretable view of how MAPFL adapts its sharing pattern across tasks.

These MAPFL-specific hyper-parameters were selected once on a small validation split and then kept *fixed across both benchmarks*. Moreover, the ablation results in Section 4.5 (e.g., forcing immediate decoupling) indicate that MAPFL is not sensitive to the precise plateau schedule, and the same configuration works well on both ISC and HAR without per-dataset retuning. Personalised encoders and heads use the same SGD optimiser as the global copy, with personalised head updates using a slightly larger head learning rate (e.g., 5×10^{-3}) for faster adaptation.

4.2.3 | *Evaluation Protocol*

For all experiments we report classification *accuracy* (ACC) and the area under the ROC curve (AUC) on held-out test data. On ISC, AUC and ACC are computed at the image level, averaging

across all clients; on HAR, metrics are computed at the window level and averaged across users.

Each configuration is run with $n = 3$ independent trials (random seeds 0, 1, 2; different client sampling orders and weight initializations). We report the mean and standard deviation over these runs. Unless stated otherwise, we select the best global model according to validation accuracy and report the corresponding test performance. All methods share exactly the same train/validation/test splits and the same number of communication rounds.

4.3 | Results on the Affective-Computing Benchmark

We first study visual sentiment analysis on ISC, where label and concept shifts dominate. Tables 2 and 3 summarise the test AUC and ACC for all methods on the 2-class and 8-class tasks, respectively. Figures 2–5 show the corresponding validation curves.

4.3.1 | 2-Class Sentiment Classification

On the binary ISC-2 task with a pre-trained MobileNetV2 encoder, all personalised and mixture-style methods provide some improvement over purely global training. Nevertheless, Table 2 and Figure 2 reveal clear structural differences.

MAPFL converges the fastest and reaches the highest validation accuracy among all methods. Its test AUC and ACC are around 88.2% and 82.5% respectively, making it the only method that simultaneously exceeds 88% AUC and 82% ACC in this setting. Strong personalised baselines such as APFL, APPLE, FedBN and FedBABU form a second performance tier: they improve over FedAvg and FedProx but plateau at noticeably lower validation curves in Figures 2 and 3.

This behaviour is consistent with MAPFL's design. By adaptively mixing global and personalised encoder parameters while keeping the classifier heads local, MAPFL can share visual features whenever clients agree (e.g., clear positive/negative images) and deviate towards client-specific encoders when subjective ratings diverge. This reduces destructive interference across clients and leads to smoother convergence than methods that either keep the encoder fully global or personalise only the head.

4.3.2 | 2-Class Training From Scratch

When MobileNetV2 is trained from scratch on ISC, the optimization landscape becomes substantially more challenging. Global and regularisation-based methods must jointly learn a feature extractor and classifier under strong subjectivity and label imbalance. Figures 4 and 5 show that several baselines quickly collapse to degenerate solutions, with validation curves saturating at low accuracy and exhibiting near-zero variance across runs.

In contrast, MAPFL still converges reliably. Its validation accuracy and AUC increase steadily throughout training and remain clearly ahead of all competitors. In Table 2, MAPFL achieves around 85.2% AUC and 80.0% ACC when trained from scratch, while many baselines either fail to improve beyond their pre-trained performance or even degrade. The gap between MAPFL and the second-best method widens compared to the pre-trained case, suggesting that encoder-level personalisation becomes even more critical when no high-quality initialisation is available: MAPFL avoids collapsing to majority predictions by rebalancing global and personalised encoders according to gradient alignment.

4.3.3 | 8-Class Emotion Recognition

On the fine-grained ISC-8 task (Table 3), the combination of ambiguous labels and highly imbalanced client-specific class distributions further stresses FL algorithms. With a pre-trained backbone, global and regularisation-based methods obtain moderate AUC but struggle to turn this into strong multi-class accuracy, and some methods show large variance across seeds.

MAPFL again stands out. It is the only method whose average performance exceeds 80% AUC and 50% ACC across pre-trained and scratch settings. Personalised and partition-based algorithms such as FedBABU, FedBN, APFL and APPLE constitute the second tier: while they outperform purely global training, they suffer from the long-tailed label distributions and tend to overfit the majority emotions. In contrast, MAPFL maintains stable performance across seeds and clearly higher AUC and ACC.

When all methods are trained from scratch on the 8-class task, the global model must serve clients with very different emotion priors and strongly skewed local label sets. Several global and regularisation-based methods saturate at low AUC and ACC, again with almost flat validation curves. MAPFL maintains a clear lead and remains numerically stable. Taken together, Tables 2 and 3 together with Figures 2–5 support the design intuition of MAPFL on affective computing: personalising at the encoder via gradient-aligned mixing, while keeping classifier heads local, offers a robust way to share global knowledge under severe concept and label shifts, for both coarse and fine-grained sentiment tasks and with or without pre-training.

4.4 | Results on the HAR Benchmark

We now turn to the HAR benchmark, where heterogeneity is mainly feature- and mixture-centric and each client contributes only 10% of its windows for training. Table 4 summarises the test AUC and ACC for all methods using the HARCNN backbone, while Figure 6 contrasts MAPFL with FedBABU through training curves and t-SNE visualizations.

4.4.1 | Overall Comparison Under Limited Labels

Despite the scarcity of labelled data, most FL algorithms reach AUC values above 93% on HAR, indicating that the six activities are objectively distinguishable once a suitable representation is learned. However, the accuracy column in Table 4 reveals notable differences in how effectively each method translates high AUC into correct per-window predictions.

Global and regularisation-based methods (FedAvg, FedProx, FedLC, FedNTD, cwFedAvg) converge to similar AUC values around 93.5% but their accuracies remain in the mid-70% range,

suggesting that they still confuse a substantial fraction of activity windows under the 10%/90% split. Personalised and mixture-style methods such as APFL, APPLE and FedDBE narrow this gap, raising accuracy to around 79%.

MAPFL further improves on this tier. It reaches an AUC of 94.33% ± 0.19 and an ACC of 81.77% ± 0.34, achieving the best performance on both metrics under the same training recipe. The absolute gain in ACC over the second-best method is modest but consistent across seeds, highlighting that MAPFL extracts more useful shared structure from scarce and heterogeneous sensor data.

A particularly striking phenomenon in Table 4 is the behaviour of FedBABU. Although FedBABU is among the strongest baselines on ISC (often ranking second to MAPFL in Tables 2 and 3), its performance on HAR collapses: the accuracy drops to about 42% with a much lower AUC. Figure 6a shows that FedBABU's validation accuracy stagnates at a low level and exhibits large fluctuations across rounds.

4.4.1.1 | *Hypothesis.* This behaviour is consistent with the dominant *feature/mixture shift* in HAR and with FedBABU's reliance on a largely shared feature extractor with predominantly *head-only* personalisation. When the shared encoder is misaligned with a client's sensor distribution (e.g., different device orientation/placement, gait style, or sensor noise), the resulting representation can be entangled across classes; under our 10% labelled setting, the client has limited supervision to compensate for representation errors by only adapting the classifier head, which can lead to unstable optimization and poor generalisation. MAPFL avoids this failure mode by allowing the encoder to deviate from the global copy through gradient-aligned mixing, while still exploiting globally shared structure.

In contrast, Figure 6c shows that MAPFL quickly reaches a high AUC and steadily increasing accuracy with a narrow uncertainty band. By allowing encoder parameters to adapt through gradient-aligned mixing while still exploiting global information, MAPFL better accommodates subject-specific feature distributions under limited labels.

4.4.2 | *Representation Analysis via t-SNE*

To further understand why FedBABU fails on HAR while MAPFL generalises well, we visualise the learned feature embeddings for the test set using t-SNE. We extract the penultimate-layer features of HARCNN and project them to 2D; each point is coloured by its ground-truth activity label. Figure 6b shows FedBABU, and Figure 6d shows MAPFL.

Under MAPFL, the six activities form compact, well-separated clusters. Dynamic activities (walking, walking upstairs/downstairs) occupy distinct regions with clear boundaries, while static activities (sitting, standing, lying) are separated but remain relatively close due to their similar motion patterns. The small intra-class variance and limited inter-class overlap are consistent with the near-perfect recognition performance in Table 4.

For FedBABU, the embedding space is far more entangled: several activity clusters overlap heavily, and many points lie in ambiguous regions where their nearest neighbours belong to multiple classes. This indicates that a rigidly shared encoder cannot reconcile the strong feature heterogeneity across subjects; local fine-tuning of heads alone cannot fix a misaligned representation. Together with the training curves in Figure 6a–c, this visual comparison illustrates that MAPFL's encoder-level personalisation is essential for handling cross-device feature shifts.

4.5 | Analysis of Personalisation Coefficients and Interpretability

A distinctive aspect of MAPFL is that each client maintains module-level personalisation coefficients $(\alpha_{b,i}, \alpha_{h,i})$ that control how much it relies on global versus local parameters for the encoder (base) and the head, respectively. These coefficients are updated by gradient-aligned rules and logged after each round. Beyond acting as trainable hyper-parameters, they provide an interpretable view of how much each client chooses to share or personalise at different network modules.

Figures 7 and 8 visualise the evolution of these coefficients on ISC (2-class) and HAR, respectively. For each dataset, panel (a) plots the mean and variance of $\alpha_{b,i}$ and $\alpha_{h,i}$ across clients over communication rounds, while panels (b) and (c) show per-client trajectories as heatmaps.

4.5.1 | *Dynamics on Visual Sentiment (ISC)*

On the ISC-2 task, Figure 7a shows that both encoder and head coefficients quickly depart from their neutral initialisation around 0.5 and stabilise at distinct operating points. The confidence bands are relatively narrow, indicating coherent behaviour across clients. The per-client heatmaps in Figure 7b,c reveal more structure: encoder coefficients display a relatively homogeneous pattern, while head coefficients exhibit more diversity, with some clients relying more heavily on local heads than others.

This pattern matches the nature of the ISC task. Visual appearance across clients is relatively similar, so encoders can share a substantial amount of information. In contrast, label semantics are subjective and vary from user to user; some annotators may be more 'positive' or 'negative' than others. MAPFL captures this by keeping encoder mixing coefficients in a moderate sharing regime while allowing head coefficients to drift more to the local side for clients whose labels deviate from the majority. In this sense, $(\alpha_{b,i}, \alpha_{h,i})$ provide a compact and interpretable summary of *how* each client negotiates between global consensus and personalised heads.

4.5.2 | *Dynamics on Human Activity Recognition (HAR)*

On HAR, Figure 8a exhibits a qualitatively different pattern. Here the encoder coefficients show larger client-to-client variation and many clients gradually move towards regimes that place more weight on local encoders, while head coefficients remain more concentrated around intermediate values. The heatmaps in Figure 8b,c confirm that encoder personalisation is more aggressive and heterogeneous than head personalisation.

This behaviour is aligned with the nature of HAR: sensor characteristics (placement, gait style, device noise) are highly client-specific, so encoders benefit from adapting strongly to each user. In contrast, the semantic boundaries between activities (e.g., walking vs. sitting) are largely shared across the population, so keeping the classifier head closer to the global model is sensible. MAPFL automatically reconfigures the degree of sharing between encoder and head when moving from concept-shift dominated ISC to feature-shift dominated HAR, without any change in code or manual retuning. The learned coefficients thus serve as an interpretable diagnostic of the underlying heterogeneity: where feature shift dominates, MAPFL chooses more local encoders; where label shift dominates, it chooses more local heads.

4.6 | Ablation Studies and Hyper-Parameter Analysis

We now investigate which components of MAPFL are most important, and how sensitive the method is to its main hyper-parameters. The focus is on the challenging ISC-8 task, where subjectivity and class imbalance are strongest, and on the ISC benchmark with MobileNetV2 trained from scratch ('mobilenet2-np') to verify that the observations hold when no external pre-training is available.

Recall that Section 3 organises MAPFL into three modules: the Dual-Path Modular Backbone (M1), the Gradient-Aligned Mixture Controller (M2), and the Plateau-Scheduled Decoupling mechanism (M3). For ablation we disentangle these modules into four key design elements:

1. *Adaptive modular mixing (M2).* Each client maintains encoder and head mixing coefficients $(\alpha_{b,i}, \alpha_{h,i})$ which are updated by gradient-aligned rules instead of being fixed in advance.
2. *Balanced head personalisation (M1+M2).* Classifier heads are always kept local and updated using both global-path and personalised-path features, so that subjective label semantics can be captured without ever averaging heads on the server.
3. *Plateau-scheduled decoupling (M3).* A client-side plateau detector monitors the personalised loss and, once improvement saturates, transitions from a coupled regime (shared coefficient) to a fully decoupled regime where encoder and head coefficients are optimised separately.
4. *Coefficient regularisation (M2).* A small quadratic penalty on α discourages extreme values and stabilises the update dynamics.

To understand the contribution of these components, we start from full MAPFL and progressively remove different pieces on ISC-8 with a *pre-trained* MobileNetV2 backbone. Table 5 summarises the most representative variants.

TABLE 5 | Ablation of MAPFL on *ISC-8* with a pre-trained MobileNetV2 backbone.

Variant	AUC (%)	ACC (%)
MAPFL (full)	**83.23**	**53.46**
No-mix (purely global encoder & head)	78.01	46.06
No-base-mix (only head mixed)	82.78	52.36
No-head-mix (only encoder mixed)	79.82	48.42
No-adapt (fixed $\alpha_{b,i}, \alpha_{h,i}$)	79.06	46.88
Force-decouple (decoupled from round 1)	82.78	52.4

Note: Best in bold, second in underline.

4.6.1 | *Encoder Versus Head Mixing*

The first three rows below full MAPFL isolate the effect of mixing at different modules. Removing *all* mixing (no-mix) drives MAPFL back to a purely global model and causes a large drop in performance: AUC falls by almost 5 points (from 83.2% to 78.4%) and ACC by over 6 points (from 53.5% to 47.3%). Disabling only head mixing while still mixing encoders (no-head-mix) is similarly detrimental, with ACC dropping by about 4.8 points. In contrast, removing encoder mixing but preserving head mixing (no-base-mix) reduces AUC and ACC only slightly ($\approx$ 0.1–0.5 points).

These comparisons highlight a key property of ISC: because client heterogeneity is dominated by subjective label semantics, carefully balancing global and personalised information at the *head* is far more important than aggressively personalising the encoder. MAPFL's design—global encoder with controllable local deviation, plus fully local heads updated from both global- and personalised-path features—matches this intuition and explains why the head-centric ablations are the most harmful.

4.6.2 | *Importance of Adaptive Coefficients*

The no-adapt variant fixes $(\alpha_{b,i}, \alpha_{h,i})$ to constants and removes the gradient-aligned controller. Compared with full MAPFL, both AUC and ACC drop by roughly 3.8–4.8 points on ISC-8. This shows that it is not enough to choose a single global sharing pattern in advance: different clients require different degrees of encoder and head personalisation, and letting α adapt to local gradients is crucial on the hardest ISC setting.

4.6.3 | *Role of Decoupling Schedule*

Finally, force-decouple disables the plateau detector and decouples encoder and head coefficients from the first round. Its performance is almost indistinguishable from full MAPFL (within 0.1–0.4 points in both metrics). This suggests that the precise trigger time of decoupling is not critical as long as encoder and head coefficients are eventually optimised separately. The plateau detector primarily serves as a robust default schedule rather than a finely tuned hyper-parameter.

4.6.3.1 | *From Pre-Trained to Scratch: Are the Trends Robust?*

To check that these conclusions are not an artefact of pre-training, we repeat the ablations on ISC-2 and ISC-8 with MobileNetV2 trained *from scratch* ('mobilenet2-np'). We keep the same default initialisation $\left(\alpha_{b,i}^{(0)}, \alpha_{h,i}^{(0)}\right) = (0.5, 0.5)$ and training recipe.

On the 8-class task with MobileNetV2-np, full MAPFL achieves around 77.4% AUC and 45.9% ACC on the validation set (the

aB0.5-aH0.5 configuration). The relative ordering of ablations mirrors the pre-trained case but with even sharper gaps:

1. Removing head mixing or all mixing (no-head-mix and no-mix) drops AUC by about 2.4 points and ACC by roughly 5.5 points, down to $\sim 75\%$ AUC and $\sim 40\%$ ACC.
2. Fixing the coefficients (no-adapt and its static variants) reduces AUC by about 1.4 points and ACC by 2–3 points.
3. In contrast, variants that preserve adaptive head mixing but alter encoder mixing or the decoupling schedule (no-base-mix and force-decouple) stay within approximately 0.5 points of full MAPFL in both AUC and ACC.

On the easier 2-class task with MobileNetV2-np, ablations lie on a broad near-optimal plateau: full MAPFL attains around 85.1% AUC and 80.3% ACC, and all variants remain within roughly 1% point. The only consistent pattern is that removing head mixing (no-head-mix and no-mix) still loses about 1–1.2 points in ACC. This is expected: coarse binary sentiment is less sensitive to the exact sharing pattern than fine-grained 8-class emotion recognition.

4.6.3.2 | *Core Takeaway.* Across both pre-trained and from-scratch settings, the ablations tell a consistent story. The main performance gains come from MAPFL's modular, gradient-aligned personalisation mechanism—embodied by the Gradient-Aligned Mixture Controller (M2) together with adaptive head mixing in the Dual-Path Modular Backbone (M1). As long as this controller is intact, the method is robust to variations in encoder mixing and in the precise schedule used by the Plateau-Scheduled Decoupling module (M3). Extreme configurations that are purely global or that fix the sharing pattern in advance perform substantially worse on the hardest ISC setting. MAPFL can therefore be viewed as a principled compromise between theory and practice: it maintains enough structure to enable cross-client transfer, yet is flexible enough to let each client negotiate its own position between global and local regimes at different network modules.

4.6.4 | *Hyper-Parameter Sensitivity*

We next study how sensitive MAPFL is to the choice of mixing and adaptation hyper-parameters in the Gradient-Aligned Mixture Controller (M2). In full MAPFL each client maintains its own $\alpha_{b,i}$ and $\alpha_{h,i}$, initialised to moderate values (e.g., 0.5) and updated adaptively via the gradient-aligned rule. Two questions are particularly important in practice: (i) how much does performance depend on the *initial* mixture $\left(\alpha_b^{(0)}, \alpha_h^{(0)}\right)$; and (ii) how much is gained by learning these coefficients at all, as opposed to using a static mixture?

4.6.4.1 | *Initial Mixing Coefficients.* To answer the first question we sweep the initial values $\left(\alpha_b^{(0)}, \alpha_h^{(0)}\right)$ over a 3×3 grid $\{0.2, 0.5, 0.8\}^2$ on ISC-2 and ISC-8 with MobileNetV2 trained from scratch ('mobilenet2-np'). In all cases the coefficients are then updated by the gradient-aligned rule.

On the 2-class task, the nine runs span AUC values between approximately 85.0% and 85.9% and ACC between 79.5% and 81.6%: the entire grid lies within about 0.9 points in AUC and 2.1 points in ACC. On the 8-class task the variation is even smaller in accuracy: AUC ranges from roughly 77.0% to 78.8%, and ACC from 45.6% to 46.6%. Settings with slightly larger head initialisation (e.g., $\alpha_h^{(0)} = 0.8$) tend to give the best numbers on both tasks, reflecting that subjective label semantics benefit from a more local head, but the effect is modest: all non-degenerate initializations end up in a narrow performance band.

These grid experiments are consistent with the coefficient dynamics visualised in Figures 7 and 8. Even though all clients start from the same neutral initialisation $\left(\alpha_b^{(0)}, \alpha_h^{(0)}\right) = (0.5, 0.5)$ in the main experiments, the trajectories quickly move to dataset-specific operating points and then fluctuate within a tight band. The fact that widely different initializations on MobileNetV2-np converge to very similar final AUC and ACC indicates that MAPFL possesses a large 'safe plateau' in the space of mixing hyper-parameters: as long as the initial coefficients are not pushed to extremes (purely global or purely local), training automatically steers them towards sensible values.

4.6.4.2 | *Value of Adaptive Coefficients.* To isolate the value of *learning* the coefficients, we turn again to the no-adapt ablation, which implements a simplified MAPFL-Static-α variant where the mixing weights are fixed to shared scalars for all clients and all rounds.

On the 2-class ISC task with MobileNetV2-np, static-α remains reasonably competitive: compared with full MAPFL, AUC changes by less than one percentage point and ACC by less than a point as well. This matches the observation that binary sentiment classification already lies on a broad plateau where many sharing patterns work. In contrast, on the more challenging 8-class task, fixing the coefficients leads to a noticeable degradation: relative to the adaptive model (initialised at (0.5, 0.5)), the static variants lose about 1.4 points in AUC and 2–3 points in ACC. Together with the strong drops observed for no-head-mix and no-mix, this shows that the gradient-aligned controller is especially important when label distributions are highly skewed and ambiguous.

From a conceptual perspective, these results reinforce the view that $\left(\alpha_{b,i}, \alpha_{h,i}\right)$ behave like trainable hyper-parameters or 'soft architecture choices'. Figure 7 shows that on ISC the head coefficients spread more across clients than the encoder coefficients, while Figure 8 shows the opposite trend on HAR, where feature shift dominates. The new hyper-parameter sweeps reveal that MAPFL does not rely on finely tuned numeric values: a wide range of initial mixtures leads to essentially the same operating regime, and adaptively learning the coefficients mainly serves to *place* each client in the right part of this regime given its own data.

Overall, MAPFL is robust to its mixing hyper-parameters and gains most from the inductive bias encoded by its modular architecture and gradient-aligned personalisation rule, rather than from delicate tuning of a single mixture weight.

5 | Conclusion

In this paper, we addressed personalised federated learning for human-centric sensing under multi-shift heterogeneity, including subjective label and concept drift in image-based sentiment analysis and feature and mixture shift in

smartphone-based human activity recognition. We proposed MAPFL, a modality-agnostic framework built on three interacting modules: a Dual-Path Modular Backbone that decouples encoder and head parameters while maintaining global and personalised copies on each client, a Gradient-Aligned Mixture Controller that adaptively interpolates between global and local parameters via module-level mixing coefficients, and a Plateau-Scheduled Decoupling mechanism that gradually transitions from a single shared coefficient to decoupled encoder and head coefficients once personalised performance plateaus.

Extensive experiments on multiple affective-computing and HAR benchmarks show that MAPFL consistently outperforms a range of federated and personalised baselines under a unified training recipe, and remains effective both with and without pre-trained backbones. MAPFL mitigates the performance collapse observed in methods that rely on a frozen shared encoder, and ablation studies confirm that combining modular encoder–head decoupling with gradient-aligned personalisation yields a favourable trade-off between accuracy, stability, and implementation simplicity. In future work, we plan to extend MAPFL to more diverse modalities (e.g., speech and physiological signals) and more challenging cross-device federated settings with partial participation and communication constraints. It would also be interesting to design more adaptive mixture controllers and decoupling schedules that react to client-specific shifts over time, and to integrate MAPFL with continual and domain-generalisation techniques to better handle evolving tasks and unseen environments.

Funding

This work was supported by Wuhan Natural Science Foundation (Grant 2024040801020236), National Natural Science Foundation of China (Grants 62225113 and 62271357), National Key Research and Development Program of China (Grant 2023YFC2705700).

Conflicts of Interest

The authors declare no conflicts of interest.

Data Availability Statement

The authors confirms that all data generated or analyzed during this study are included in this published article.

References

1. H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Aguera y Arcas, "Communication-Efficient Learning of Deep Networks From Decentralized Data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS)* (PMLR, 2017), 1273–1282.

2. M. Ye, X. Fang, B. Du, P. C. Yuen, and D. Tao, "Heterogeneous Federated Learning: State-of-the-Art and Research Challenges," *ACM Computing Surveys* 56, no. 3 (2023): 1–44, https://doi.org/10.1145/3625558.

3. C. Dhasaratha, M. K. Hasan, S. Islam, et al., "Data Privacy Model Using Blockchain Reinforcement Federated Learning Approach for Scalable Internet of Medical Things," *CAAI Transactions on Intelligence Technology* 9, no. 1 (2024): 130–146, https://doi.org/10.1049/cit2.12287.

4. J. Xu, B. S. Glicksberg, C. Su, P. Walker, J. Bian, and F. Wang, "Federated Learning for Healthcare Informatics," *Journal of Healthcare Informatics Research* 5 (2021): 1–19, https://doi.org/10.1007/s41666-020-00082-4.

5. C. Liu, Y. Luo, Y. Xu, and B. Du, "Foundation Models Matter: Federated Learning for Multi-Center Tuberculosis Diagnosis via Adaptive Regularization and Model-Contrastive Learning," *World Wide Web* 27, no. 3 (2024): 30, https://doi.org/10.1007/s11280-024-01266-3.

6. J. Kang, Z. Xiong, D. Niyato, S. Xie, J. Zhang, and M. Guizani, "Reliable Federated Learning for Mobile Networks," *IEEE Wireless Communications* 27, no. 2 (2020): 72–80, https://doi.org/10.1109/mwc.001.1900119.

7. L. Tu, X. Ouyang, J. Zhou, Y. He, and G. Xing, "FedDL: Federated Learning via Dynamic Layer Sharing for Human Activity Recognition," in *Proceedings of the 19th ACM Conference on Embedded Networked Sensor Systems (SenSys)* (2021), 15–28.

8. W. Huang, M. Ye, Z. Shi, et al., "Federated Learning for Generalization, Robustness, Fairness: A Survey and Benchmark," *IEEE Transactions on Pattern Analysis and Machine Intelligence* 46, no. 12 (2024): 9387–9406, https://doi.org/10.1109/tpami.2024.3418862.

9. C. Liu, Z. Wang, Y. Xu, and B. Du, "CAFL: Conditional Attention Federated Learning for Image Emotion Analysis," *Data Science and Engineering* (2025).

10. Y. Yang, L. Deng, S. Xu, Y. Wang, and D. Yan, "Personalized Image Aesthetics Assessment With Rich Attributes," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), 19861–19869.

11. D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A Public Domain Dataset for Human Activity Recognition Using Smartphones," in *Proceedings of the 21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)* (2013), 437–442.

12. A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards Personalized Federated Learning," *IEEE Transactions on Neural Networks and Learning Systems* 34, no. 12 (2023): 9587–9603, https://doi.org/10.1109/tnnls.2022.3160699.

13. T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, "Federated Optimization in Heterogeneous Networks," *Proceedings of Machine Learning and Systems (MLSys)* 2 (2020): 429–450, https://doi.org/10.48550/arXiv.1812.06127.

14. S. P. Karimireddy, S. Kale, M. Mohri, S. Reddi, S. U. Stich, and A. T. Suresh, "SCAFFOLD: Stochastic Controlled Averaging for Federated Learning," in *Proceedings of the 37th International Conference on Machine Learning (ICML)* (PMLR, 2020), 5132–5143.

15. D. A. E. Acar, Y. Zhao, R. Matas, M. Mattina, P. Whatmough, and V. Saligrama, "Federated Learning Based on Dynamic Regularization," in *International Conference on Learning Representations (ICLR)* (2021).

16. A. Fallah, A. Mokhtari, and A. Ozdaglar, "Personalized Federated Learning With Theoretical Guarantees: A Model-Agnostic Meta-Learning Approach," *Advances in Neural Information Processing Systems* 33 (2020): 3557–3568, https://doi.org/10.5555/3495724.3496024.

17. V. Smith, C.-K. Chiang, M. Sanjabi, and A. S. Talwalkar, "Federated Multi-Task Learning," *Advances in Neural Information Processing Systems* 30 (2017), https://doi.org/10.5555/3294996.3295196.

18. M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, "Federated Learning With Personalization Layers," *Preprint arXiv: 1912.00818* (2019).

19. S. Pieri, C. Rusu, J. M. Tomczak, and J. van Gemert, "Handling Data Heterogeneity via Architectural Design for Federated Visual Recognition," *Advances in Neural Information Processing Systems* 36 (2023), https://doi.org/10.52202/075280-0180.

20. Z. Qu, R. Wang, M. Li, H. Wang, J. Gao, and X. Li, "Rethinking Architecture Design for Tackling Data Heterogeneity in Federated Learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), 10065–10075.

21. Y. Deng, M. M. Kamani, and M. Mahdavi, "Adaptive Personalized Federated Learning," *Preprint arXiv:2003.13461* (2020).

22. J. Zhang, Y. Hua, H. Wang, et al., "FedALA: Adaptive Local Aggregation for Personalized Federated Learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 37, no. 9 (2023), 11237–11245, https://doi.org/10.1609/aaai.v37i9.26330.

23. J. Zhang, Y. Hua, H. Wang, and T. Song, "GPFL: Simultaneously Learning Generic and Personalized Feature Information for Personalized Federated Learning," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2023), 5041–5051.

24. Y. Tan, G. Long, L. Liu, et al., "FedProto: Federated Prototype Learning Across Heterogeneous Clients," in *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 36, no. 8 (2022), 8432–8440, https://doi.org/10.1609/aaai.v36i8.20819.

25. F. Sattler, K.-R. Müller, and T. Wiegand, "Clustered Federated Learning: Model-Agnostic Distributed Multitask Optimization Under Privacy Constraints," *IEEE Transactions on Neural Networks and Learning Systems* 32, no. 8 (2021): 3710–3722, https://doi.org/10.1109/tnnls.2020.3015958.

26. J. Zhang, Y. Hua, H. Wang, et al., "Eliminating Domain Bias for Federated Learning in Representation Space," *Advances in Neural Information Processing Systems* 36 (2023): 45846–45865, https://doi.org/10.52202/075280-0625.

27. W. Huang, M. Ye, and B. Du, "Learn From Others and Be Yourself in Heterogeneous Federated Learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2022), 10143–10153.

28. S. Guo, X. Wang, S. Long, H. Liu, H. Liu, and T. H. Sam, "A Federated Learning Scheme Meets Dynamic Differential Privacy," *CAAI Transactions on Intelligence Technology* 8, no. 3 (2023): 1087–1100, https://doi.org/10.1049/cit2.12187.

29. H. Lyu, Y. Zhai, Z. Liu, H. Liu, S. Guo, and X. He, "Federated Learning Privacy Incentives: Reverse Auctions and Negotiations," *CAAI Transactions on Intelligence Technology* 8, no. 4 (2023): 1538–1557, https://doi.org/10.1049/cit2.12190.

30. J. Jiang and L. Hu, "Decentralised Federated Learning With Adaptive Partial Gradient Aggregation," *CAAI Transactions on Intelligence Technology* 5, no. 4 (2020): 280–288, https://doi.org/10.1049/trit.2020.0082.

31. Y. Hui, Y. Wang, Y. Chen, R. Zhang, and M. Li, "Ada-FFL: Adaptive Computing Fairness Federated Learning," *CAAI Transactions on Intelligence Technology* 9, no. 3 (2024): 573–584, https://doi.org/10.1049/cit2.12232.

32. C. Zhou, J. Liu, J. Jia, et al., "Efficient Device Scheduling With Multi-Job Federated Learning," *Proceedings of the AAAI Conference on Artificial Intelligence* 36, no. 9 (2022): 9971–9979, https://doi.org/10.1609/aaai.v36i9.21235.

33. Y. Zhang, Y. Hu, X. Gao, "An Embedded Vertical-Federated Feature Selection Algorithm Based on Particle Swarm Optimisation," *CAAI Transactions on Intelligence Technology* 8, no. 3 (2023): 734–754, https://doi.org/10.1049/cit2.12122.

34. M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted Residuals and Linear Bottlenecks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2018), 4510–4520.

35. J. Zhang, Y. Liu, Y. Hua, et al., "PFLlib: A Beginner-Friendly and Comprehensive Personalized Federated Learning Library and Benchmark," *Journal of Machine Learning Research* 26, no. 50 (2025): 1–10, https://doi.org/10.48550/arXiv.2312.04992.

36. J. Machajdik and A. Hanbury, "Affective Image Classification Using Features Inspired by Psychology and Art Theory," in *Proceedings of the 18th ACM International Conference on Multimedia* (2010), 83–92.

37. X. Li, M. Jiang, X. Zhang, M. Kamp, and Q. Dou, "FedBN: Federated Learning on Non-IID Features via Local Batch Normalization," in *International Conference on Learning Representations (ICLR)* (2021).

38. Q. Li, B. He, and D. Song, "Model-Contrastive Federated Learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2021), 10713–10722.

39. G. Lee, M. Jeong, Y. Shin, S. Bae, and S.-Y. Yun, "Preservation of the Global Knowledge by Not-True Distillation in Federated Learning," *Advances in Neural Information Processing Systems* 35 (2022): 38461–38474, https://doi.org/10.52202/068431-2787.

40. J. Zhang, Z. Li, B. Li, et al., "Federated Learning With Label Distribution Skew via Logits Calibration," in *Proceedings of the 39th International Conference on Machine Learning (ICML)* (PMLR, 2022), 26311–26329.

41. J. Oh, S. Kim, and S.-Y. Yun, "FedBABU: Towards Enhanced Representation for Federated Image Classification," in *International Conference on Learning Representations (ICLR)* (2022).

42. G. Lee and D. Choi, "Class-Wise Federated Averaging for Efficient Personalization," in *Proceedings of the IEEE/CVF International Conference on Computer Vision* (2025), 1773–1782.

43. H. Wang, Z. Kaplan, D. Niu, and B. Li, "Optimizing Federated Learning on Non-IID Data With Reinforcement Learning," in *IEEE INFOCOM 2020—IEEE Conference on Computer Communications* (IEEE, 2020), 1698–1707.

44. T. Li, S. Hu, A. Beirami, and V. Smith, "Ditto: Fair and Robust Federated Learning Through Personalization," in *Proceedings of the 38th International Conference on Machine Learning (ICML)* (PMLR, 2021), 6357–6368.

45. X. Yang, W. Huang, and M. Ye, "FedAS: Bridging Inconsistency in Personalized Federated Learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2024), 11986–11995.

46. X. Fang, M. Ye, and X. Yang, "Robust Heterogeneous Federated Learning Under Data Corruption," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)* (2023), 5020–5030.

47. W. Huang, M. Ye, Z. Shi, H. Li, and B. Du, "Rethinking Federated Learning With Domain Shift: A Prototype View," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2023), 16311–16321.

48. J. Tang, Y. Xue, Z. Wang, et al., "Bayesian Estimation-Based Sentiment Word Embedding Model for Sentiment Analysis," *CAAI Transactions on Intelligence Technology* 7, no. 2 (2022): 144–155, https://doi.org/10.1049/cit2.12037.

49. X. Zhang, D. Huang, H. Li, Y. Zhang, Y. Xia, and J. Liu, "Self-Training Maximum Classifier Discrepancy for EEG Emotion Recognition," *CAAI Transactions on Intelligence Technology* 8, no. 4 (2023): 1480–1491, https://doi.org/10.1049/cit2.12174.

50. J. Luo and X. Wu, "Adapt to Adaptation: Learning Personalization for Cross-Silo Federated Learning," in *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)* (2022), 2166–2173.